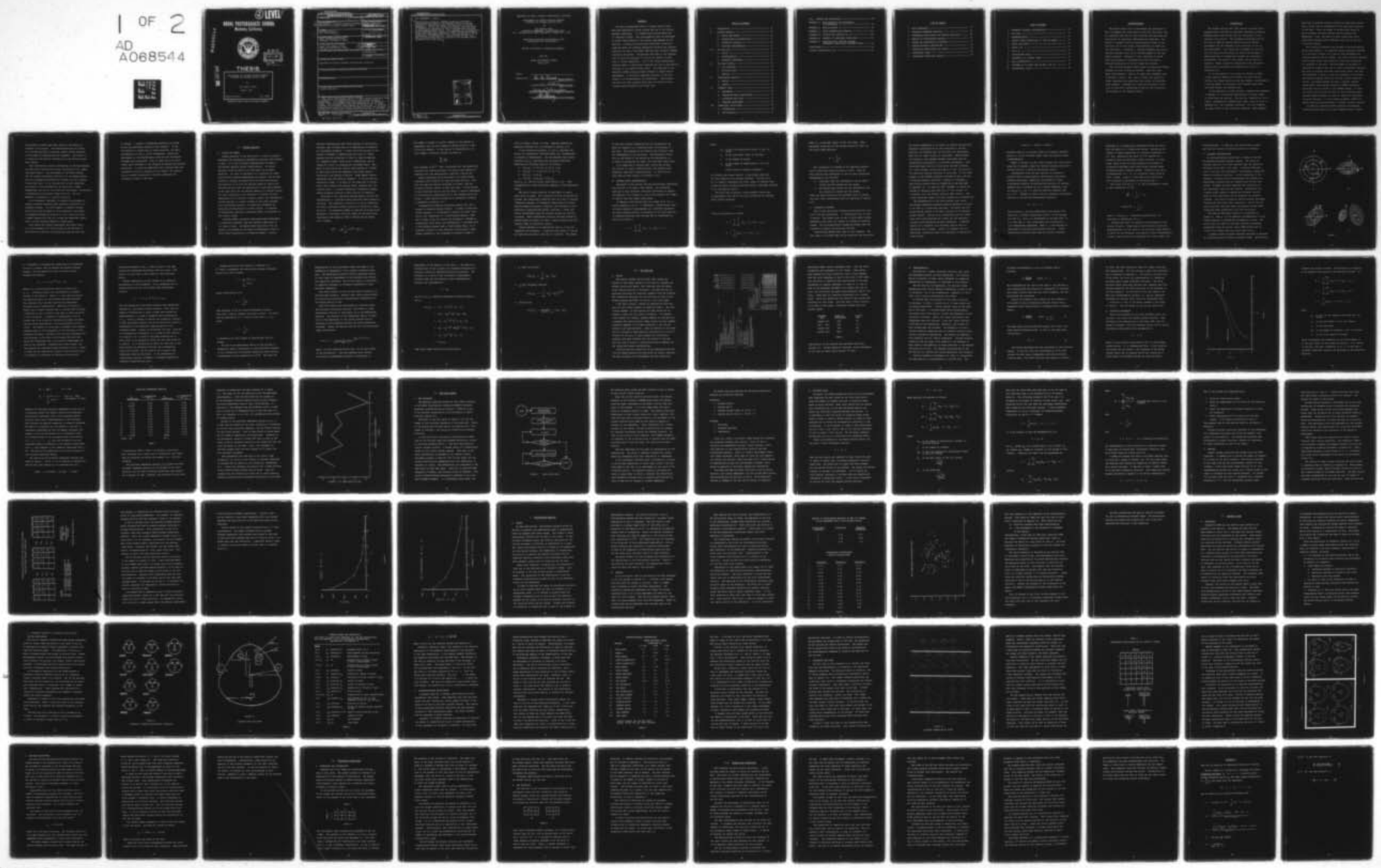
AD-A068 544 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 12/1
DEVELOPMENT OF CLUSTER ANALYSIS METHODS SUITABLE FOR STUDENT OP--ETC(U)
MAR 79 J W AIKEN

UNCLASSIFIED NL

1 OF 2
AD
A068544



NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DEVELOPMENT OF CLUSTER ANALYSIS METHODS
SUITABLE FOR STUDENT OPINION DATA

by

Joel Weston Aiken

March 1979

Thesis Advisor: R.R. Read

Approved for public release; distributior unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Development of Cluster Analysis Methods Suitable for Student Opinion Data.		5. TYPE OF REPORT & PERIOD COVERED (9) Master's Thesis, March 1979
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Joel Weston Aiken		8. CONTRACT OR GRANT NUMBER(s) (12) 152 p.
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE (11) Marc[redacted]979
		13. NUMBER OF PAGES 151
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Cluster Analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Naval Postgraduate School's Student Opinion Form data were subjected to study through the use of two cluster analysis techniques: (1) K-MEANS partitioning method and (2) Chernoff's FACES. Much developmental work was performed to tailor these methods to the special requirements of the data set. A thorough multivariate statistical review provided the basis for choosing optimality criteria and distance functions for use in the MIKCA

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251 450

#20 (ABSTRACT) - Cont'd

(Multivariate Iterative K-MEANS Clustering Algorithm). Alterations were made to the computer code to allow the analysis to include the effect of class size on cluster membership. Use of the linear discriminant function aided in identifying variables for use in constructing features of the computer-drawn faces. This approach to the Chernoff's FACES technique shows promise but needs further development. A principal components analysis of the data showed it to be essentially one dimensional. Partitioning the data into four clusters shows that the scoring of the courses varies inversely with class size.

Approved for public release; distribution unlimited.

Development of Cluster Analysis Methods Suitable for Student Opinion Data

by

Joel Weston Aiken
Lieutenant Commander, United States Navy
B.S., University of North Carolina at Chapel Hill, 1969

Submitted in partial fulfillment of the
requiremnts for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1979

Author ______________________

Approved by: R R Read
Thesis Advisor

Second Reader

Chairman, Department of Operations Research

Dean of Information and Policy Sciences

ABSTRACT

The Naval Postgraduate School's Student Opinion Form data were subjected to study through the use of two cluster analysis techniques: (1) K-MEANS partitioning method and (2) Chernoff's FACES. Much developmental work was performed to tailor these methods to the special requirements of the data set. A thorough multivariate statistical review provided the basis for choosing optimality criteria and distance functions for use in the MIKCA (Multivariate Iterative K-MEANS Clustering Algorithm). Alterations were made to the computer code to allow the analysis to include the effect of class size on cluster membership. Use of the linear discriminant function aided in identifying variables for use in constructing features of the computer-drawn faces. This approach to the Chernoff's FACES technique shows promise but needs further development. A principal components analysis of the data showed it to be essentially one dimensional. Partitioning the data into four clusters shows that the scoring of the courses varies inversely with class size.

TABLE OF CONTENTS

<u>LIST OF TABLES</u>

LIST OF FIGURES

ACKNOWLEDGMENTS

The author would like to acknowledge the assistance of Mrs. Pat Meadows who organized the data from card decks into an accessible data disc so that retrieval was made easy and the privacy of individual instructors was respected. Professor G. Lindsay's work in scale analysis on part of the data was used to help select transformations to stabilize the variability. Professor J. Hartman suggested the sum of squares technique which led to the development of the comparison measure. Professor D. Kirk (Chairman of the EE Dept) participated in clustering the faces and gave a detailed description of how he formed the clusters.

The following students participated as judges and offered comments on their clustering methods: LCDR John Scott Redd, LCDR Raymond J. Morris, LT James Kevin McDermott and LT Howard S. Hilley. Mrs. Joel W. Aiken also acted as a judge, assisted with typing the first draft, and provided moral support. Professor R.R. Read gave willingly of his time in providing a background in multivariate statistics and guidance in the research effort.

I. INTRODUCTION

The Student Opinion Form (SOF) used at the Naval Postgraduate School provides an organized information gathering mechanism about each course (and its instructor) as perceived by the students. The information obtained from the SOF data is used for administrative review of faculty performance and for feedback to the instructor to aid in self-development. The former use is hampered by the fact that the data are multivariate in nature and represent a complicated set of interactions between the instructor's performance, the nature of the course, and the group of students. There is need for methodology which can disentangle those interactions and summarize the data in a meaningful way.

It is the purpose of this thesis to develop suitable cluster analysis methods for studying the data and discovering any hidden structure they may possess. Concurrently, a certain amount of exploratory data analysis took place, and those results are reported also.

At the completion of every quarter, students are requested to respond to a 16-item SOF questionnaire for each course in which they are enrolled. The data are viewed as an n by p matrix, representing n observations (SOFs), each of which is measured on p (16) different variables. For this research the mean vector of each course was computed. Then attempts

were made to discover natural clusters of these mean vectors which in turn can be interpreted as the underlying structure in the data. Since the number of students per course is quite variable, the mean vectors are not equally well determined. Also, the matrix of mean vectors may have a covariance structure quite different from that of the full n by p data matrix.

The clustering objective was pursued by two multivariate statistical methods: one computer-graphic technique referred to as Chernoff's FACES, and a second, more mathematically oriented approach called K-MEANS. The former produces computer-drawn cartoon faces, the features of which are controlled by variables in the data. The assignment of variables to features was aided by the use of linear discriminant analysis. One face is produced from each course mean vector, and then the researcher is able to study the appearance of the faces and cluster together those that display similar characteristics. The second method utilizes a computer program called MIKCA (Multivariate Iterative K-MEANS Clustering Algorithm) which is based on the K-MEANS method. It forms an initial partition of the data and then transfers observations between clusters in order to improve an optimality criterion function. In this iterative manner, MIKCA ultimately stabilizes and provides an "optimal" cluster solution.

In addition a modified MIKCA technique was employed. Alterations were made to the basic computer code to enable

the program to weight each mean vector by the number of students in the course. This modification may be likened to a one-way Analysis of Variance (ANOVA) having unbalance in the number of observations per treatment. The result is to stabilize the relative variability of the various course mean vectors.

Most multivariate analysis methodology is derived assuming the data have a multivariate normal distribution with common covariance matrix. The performance of the MIKCA program and the linear discriminant analysis will not depend greatly upon this assumption provided the clusters are well defined. On the other hand, if the clusters are not well separated, the results of the programs will be sensitive to these assumptions, and this is the condition anticipated. Accordingly, a transformation was sought toward this end. The one selected is essentially a logistic function.

It is frequently necessary to compare the agreement of cluster solutions produced under different conditons or by different methods. For this purpose, a computer program was written which provides an ad hoc measure of the amount of agreement between the results of two or more solutions. A number between zero and one, called the comparison coefficient, is the resulting measure of association.

This thesis was largely exploratory and should serve as a firm foundation for future study of the SOF data in particular and similarly structured multivariate data sets

in general. A number of unexpected questions are raised during the exploratory phases of this research. It was not possible to answer many of these questions, and their consideration is left to other researchers. During the development of the methodologies, some new and challenging problems were encountered. Many of these had to be given rather short treatment in the interest of meeting the original objectives. It should be emphasized that although some very interesting facts are revealed in this thesis, the results are by no means considered to describe completely the information hidden in the data.

II. CLUSTER ANALYSIS

A. ORIGIN AND THEORY

Cluster analysis is the name given to a body of diverse techniques for discovering taxonomical structure within bodies of data. It is one of several methodologies included in the broader category called classification. In cluster analysis little or nothing is known about the category structure. All that is available is a collection of observations whose category memberships are unknown, and one must discover a category structure which fits the observations. The objective is to find the natural groups by sorting the observations such that the association is high among members of the same group and low between members of different groups. The great challenge to the researcher is finding the most appropriate way of defining "natural groups" and "association." Cluster analysis is closely related to and often confused with discriminant analysis, a statistical procedure for assigning new observations to known groups. In contrast to discriminant analysis, clustering refers to discovery of the initial groups.

Although modern clustering techniques began development in biological taxonomy, they are generally applicable to all types of data. Any method which partitions a set of objects into subsets on the basis of measurements taken on every object qualifies as a clustering method. Cluster

analysis techniques are most often applied in multivariate settings, that is where each of n observations is measured on p different variables. A clear intuitive picture of the concept is helpful in appreciating the value of cluster analysis and the situations to which it might be applied. In a geometric sense, every object (observation) may be viewed as a point in p-dimensional Euclidean space. This swarm of data points may contain dense regions or "clouds" of data points which are separable from other regions containing a low density of points. These denser regions constitute what are known as clusters. In the one and two dimensional cases, it is easy for the human eye to quickly detect the clusters from scatter plots, assuming that the clusters exist. In higher dimensions, clustering attempts become extremely difficult without the aid of computers.

Solutions to the clustering problem usually involve the determination of a partition which satisfies some optimality criterion. The optimality criterion is a way of measuring how good a particular cluster solution is relative to other solutions. An astounding number of possible solutions exist. Reference 1 describes a Stirling number of the second kind representing the number of ways n objects may be sorted into m groups.

$$S_n^{(m)} = \frac{1}{m!} \sum_{k=0}^{m} (-1)^{m-k} \binom{m}{k} k^n$$

The number of groups is usually unknown so the problem is compounded, and the total number of possibilities is a sum of Stirling numbers. In the case of 25 observations, the total number of possible cluster solutions is

$$\sum_{j=1}^{25} S_{25}^{(j)}$$

which exceeds 4×10^{18}. This illustrates that the enumerative technique for finding solutions can require huge amounts of computer time, and there exists a need for a better way.

Modern techniques allow solutions to be found without evaluating the criterion for each and every solution. However the need for ranking solutions is evident, and the criterion function serves to meet this need. A wide variety of such functions exists, and the choice is usually determined by the particular characteristics of the research being conducted. A more detailed discussion of optimality criteria is presented in Section II.C.

Mathematical clustering techniques usually call for a concept of distance between objects. In order to solve the cluster problem, it is desirable to define the terms "similarity" and "difference" in a quantitative fashion. What does it mean to say two objects are different? Perhaps an investigator would assign two observations to the same group if the distance between them is sufficiently small, or to different clusters if this distance is sufficiently large. Common reference to the closeness of objects is made in

units of length, weight, or time. Numerous methods for measuring distance will be discussed in Section II.D.

In the following discussion, X_i and X_j represent two points in p-dimensional Euclidean space (E_p) corresponding to objects or observations. Any non-negative real-valued function $D(X_i,X_j)$ satisfying the following conditions qualifies as a distance function (or metric).

a. $D(X_i,X_j) \geq 0$ for all X_i and X_j in E_p

b. $D(X_i,X_j) = 0$ if and only if $X_i = X_j$

c. $D(X_i,X_j) = D(X_j,X_i)$

d. $D(X_i,X_j) \leq D(X_i,X_k) + D(X_k,X_j)$

where X_i, X_j, and X_k are any three points in E_p. Later discussions will place particular emphasis on the Mahalanobis metric.

The use of cluster analysis is applicable in nearly every field of study. The literature is both voluminous and diverse, the terminology differing from one field to another. "Numerical taxonomy" is frequently substituted for cluster analysis among biologists, botanists, and ecologists, while some social scientists may prefer "typology." Other frequently encountered terms are pattern recognition and partitioning. While discriminant analysis has been studied by statisticians for nearly 45 years, cluster analysis has only recently come to statistical notice.

Cluster analysis is an exploratory device, a tool for suggestion and discovery. A question often asked is "How do you know when you have a good set of clusters?" The answer

is that the clusters themselves are not interesting; the point of interest is in inference about the structure of the data. The clusters do not explain the structure; they are consequences of the structure. The explanatory structure is the object of the search and its description is in terms of principles and ideas, not individual data units.

It is important to realize that a given set of data may contain no "right" classification, but possibly many different, meaningful classifications. It could be the case that the data contain no clusters at all.

B. SCATTER MATRIX DECOMPOSITION

Described in this section are the multivariate terminology and notation to be used on this thesis. The literature contains as many different notational structures as authors. The emphasis is on simplicity, while also exposing the reader to some of the more common terminology.

In general, multivariate data are viewed as an n by p matrix referred to as X. It represents n observations, each of which consists of measurements on p different variables. The cross products matrix is analogous to the univariate sum of squared deviations from the mean and is represented by the p by p matrix T.

$$T = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - x_{..})(x_{ij} - x_{..})'$$

where

x_{ij} is the j-th observation vector in the i-th group.

$x_{..}$ is the grand mean vector of the data.

g is the number of groups.

n_i is the number of observations in the i-th group.

Prime notation indicates transpose.

All vectors are column vectors. Cross product matrices are also referred to as scatter matrices. Division of T by n-1 (where n represents the total number of observations) yields the total variance-covariance matrix, sometimes referred to as a dispersion matrix.

The total sum of squares (cross products) matrix may be expressed as the sum of the within-group and the between-group scatter matrices:

$$T = W + B$$

W and B are defined as follows:

$$W = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (x_{ij} - x_{i.})(x_{ij} - x_{i.})'$$

$$B = \sum_{i=1}^{g} n_i (x_{i.} - x_{..})(x_{i.} - x_{..})'$$

where $x_{i.}$ is the mean vector of the i-th group. Each individual group has its own scatter matrix W_i, and W is the sum of these matrices:

$$W = \sum_{i=1}^{g} W_i$$

This discussion is intended to be completely general, with no particular group structure in mind. Later we shall explore the differences in the two group structures represented by the SOF data.

(1) Individual SOFs are considered to be the observations and the courses are the groups.

(2) The course mean vectors are the observations and the clusters of courses are the groups.

These two group structures are different ways of viewing the data; their relationship shall be explained in Section IV.B.

C. OPTIMALITY CRITERIA

Most of the well known clustering techniques fall into one of two main categories: (1) hierarchical and (2) partitioning. The former class is one in which every cluster obtained at any stage is a merger of clusters at previous stages. The non-hierarchical procedures however form new clusters by lumping and splitting old ones.

Partitioning methods were used in this research. The main idea is to choose some initial partition and then alter

the cluster membership in an effort to improve the partition. Different interpretations of what constitute a "better" partition and numerous ways of achieving this improvement have led to a great variety of algorithms. These methods are related to the steepest descent algorithms used for unconstrained optimization in nonlinear programming. Such algorithms begin with an initial point and then converge to a local optimum, moving one step at a time, the value of the objective function improving at each step. A well known example is the ISODATA procedure developed by Ball and Hall at Stanford Research Institute. Chapter IV discusses a partitioning method known as K-MEANS which was developed by MacQueen [2]. He uses the term "K-MEANS" to denote the process of assigning each data unit to that cluster (of k clusters) with the nearest centroid (mean vector). The cluster centroids change with each transfer of an observation.

The decomposition of the total scatter into within and between components suggests possible optimality criteria to be used in a clustering algorithm. One would like the within-groups scatter to be small relative to the between-groups scatter. Various trial clusterings could be formed using the W and B matrices as a basis for the optimaltiy criteria which determine the best clustering. A possible choice for a criterion is to minimize trace W over all partitions into g groups. SInce T is constant over all partitions, minimizing trace W is equivalent to maximizing trace B since

$$\text{trace } T = \text{trace } W + \text{trace } B$$

Although trace W is invariant under an orthogonal transformation, it is not invariant under other non-singular linear transformations.

McRae [3] points out that trace W equals the total within group sum of squares, hence the "minimum variance partition" cluster solution is found by minimizing trace W.

Considerable study has been devoted to alternative criteria such as those based on multivariate statistical analysis techniques, especially the methods of linear discriminant analysis and multivariate analysis of variance. Assuming the p variables are not linearly dependent, then as long as $p \leq n-g$, W is positive definite symmetric and so is W^{-1}. Attempts to make B and W as different as possible lead one to solving the determinantal equation:

$$|B - \lambda W| = 0$$

The solutions λ_i are the eigenvalues of the matrix $W^{-1}B$. There are t non-zero eigenvalues, where t is the minimum of p and g-1. This is a consequence of the fact that, if g is less than p, the g group means are contained in a (g-1)-dimensional hyperplane. When $g = 2$ the analysis is equivalent to two-group discriminant analysis. Linear discriminant analysis would take the vectors originally

described in a p-dimensional coordinate system and transform the basis to a t-dimensional system. Maximizing the largest of these eigenvalues is a criterion suggested by S.N. Roy. Maximizing the trace of $W^{-1}B$, however is a criterion known as Hotelling's trace criterion. In both cases, large values for these statistics are sought in clustering algorithms since large values indicate large differences among (between) groups. Minimizing the ratio of determinants $|W| \div |T|$ is a criterion widely known as Wilks' lambda. Since T is the same for all partitions, this criterion is equivalent to minimizing det W.

Both trace $W^{-1}B$ and $|T| \div |W|$ may be expressed in terms of the eigenvalues of $W^{-1}B$.

$$\frac{|T|}{|W|} = \prod_{i=1}^{t} (1 + \lambda_i)$$

$$\text{trace } W^{-1}B = \sum_{i=1}^{t} \lambda_i$$

where $t = \min(p, g-1)$. Therefore minimizing det W is equivalent to maximizing $\pi(1+\lambda_i)$.

Friedman and Rubin [4] describe the advantages of the various criteria. Those based on multivariate statistical considerations (all but trace W) are invariant under changes in scale for the variables (non-singular linear transformation). In fact, they are the only invariants for W and B under such

transformations. In addition, the multivariate criteria may take into account covariation among the variables.

D. DISTANCE CONSIDERATIONS

As indicated earlier there exist a number of choices for measuring distance between objects. The choice of distance function if no less important than the choice of variables to be used in the study. A serious difficulty lies in the fact that knowledge of the clusters changes the choice of distance functions. In the computation of the distance, a variable which distinguishes well between two established clusters might be weighted more heavily than others. Friedman and Rubin describe this difficulty as the "bootstrap" nature of the problem. Knowledge of the clusters would suggest an appropriate distance function which in turn would allow one to determine the original clusters. The trace W criterion implies ordinary Euclidean distance and thus hides this circularity. Use of the criteria which are invariant under non-singular linear transformations deals effectively with this circularity.

The familiar Euclidean distance is illustrated in figure 1a. When $p = 2$ the geometric interpretation of this measure amounts to determining distances by circles. Two points such as A and B on the same circle are considered equidistant from the origin, while other points such as C and D are further from the origin than A and B.

A general class of squared distance functions is provided by utilizing positive definite quadratic forms. Specifically,

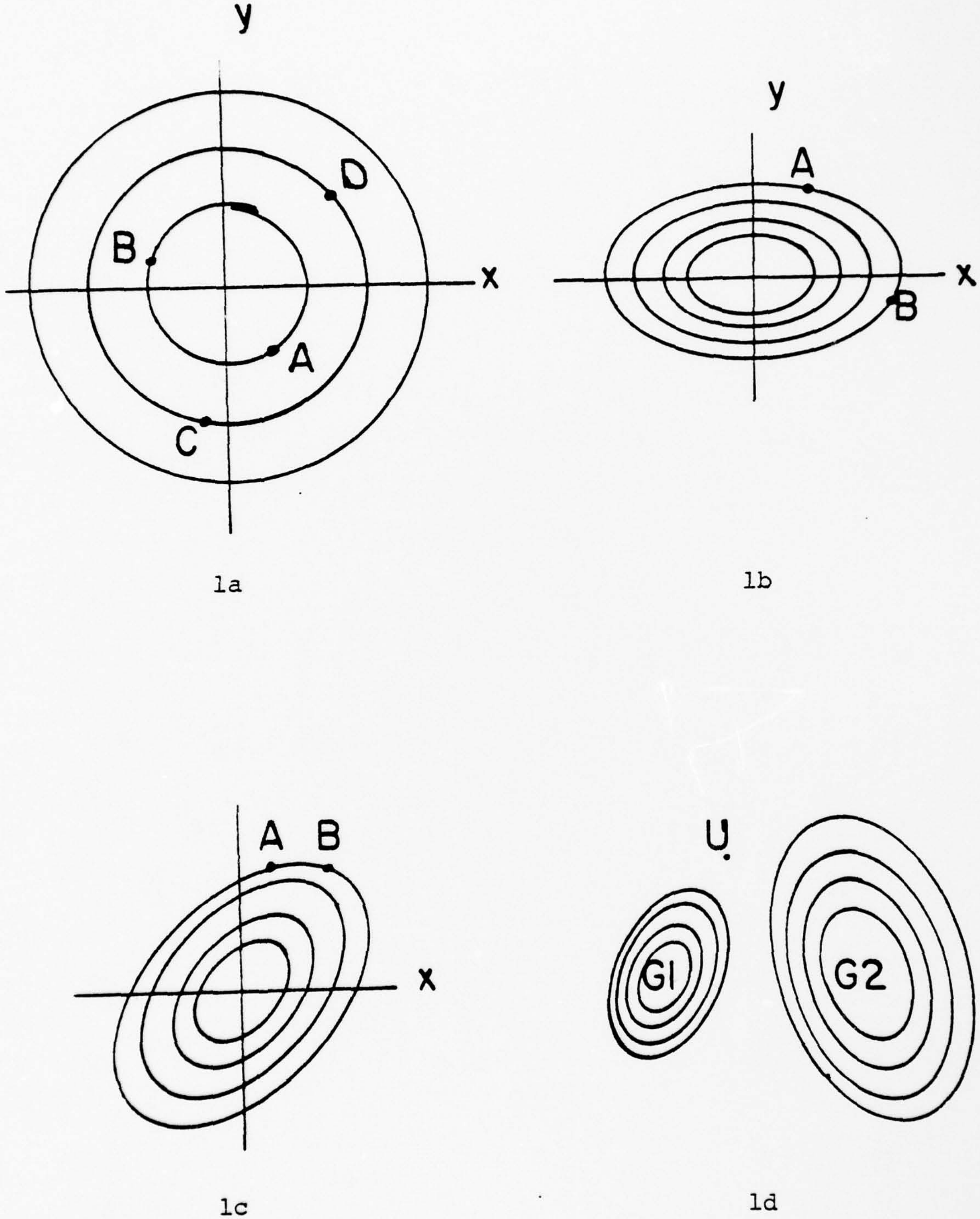


FIGURE 1

if β represents a p-dimensional observation to be assigned to one of s groups, then to measure the squared distance between β and the centroid of the i-th group one may consider the function

$$D_i = (\beta - x_{i.})^T M (\beta - x_{i.}) \qquad (1)$$

where M is a positive definite matrix to ensure that $D_i \geq 0$. Different metrics are represented by different choices of the matrix M. When M = I (the identity matrix) the resulting metric is the standard Euclidean distance. The variance within the data may make the unweighted Euclidean metric inappropriate. Referring to figure 1b where x has a larger variance than y, one may wish to weight a deviation in the x direction less than an equal deviation in the y direction. A method for accomplishing this is through use of an elliptical (weighted Euclidean) distance function which makes points A and B equidistant from the origin. The matrix M in this case is diagonal with diagonal elements equal to the reciprocals of the variances of the different variables. Insofar as the variance represents the true structure in the data, this distance function will adjust for differences due to the scale of measurement of each of the variables. Extending this idea further, one may consider the covariance among variables as well. Figure 1c shows how the axes may be tilted so that the major axis is oriented in a direction of reflecting the positive

correlation between x and y. Again, points on the same ellipse are considered equidistant from the origin. The matrix M in this case is the inverse of the covariance matrix.

Further examination of this concept is an important consideration in this research. If C_i represents the co-variance matrix of the i-th cluster then the distance function

$$D_i = (\beta - x_{i.})^T C_i^{-1} (\beta - x_{i.})$$

uses the appropriate covariance structure when determining distance to a particular cluster centroid. Note that the number of observations in every cluster must exceed the dimensionality p in order to preserve the nonsingularity of C_i. Since C_i changes to reflect the dispersion internal to each particular cluster, the use of this metric exploits differences in the dispersion characteristics of the different groups. Figure 1d illustrates the idea. Note how a new observation (denoted by u) is closer to the centroid of group one (G1) in terms of Euclidean distance but is more likely to be assigned to group two (G2) when using the C_i matrix. It is instructive to point out here that if one were looking for boundaries dividing the p-dimensional space into regions, one for each of the g groups, such boundaries would be non-linear. In the performance of discriminant analysis, Eisenbeis [5] suggests appropriate quadratic classification rules.

Another choice for the M matrix in equation 1 is C^{-1} where C represents the pooled within groups covariance matrix of all the clusters.

$$C = \frac{1}{\sum_{i=1}^{g} (n_i - 1)} W$$

Recall from Section II.B:

$$W = \sum_{k=1}^{g} W_k$$

This distance is the well known Mahalanobis distance. Note that C does not change from group to group. To ensure the non-singularity of C it must be true that $p \leq (n-g)$ where

$$n = \sum_{i=1}^{g} n_i$$

n represents the total number of observations over all groups.

The use of the Mahalanobis metric in the original p-dimensional space is equivalent to using Euclidean distance in the t-dimensional discriminant space with basis vectors corresponding to the eigenvectors of $W^{-1}B$. Note that the

determination of the discriminant space was based on the assumption of homogeneity of the cluster covariance structure. The Mahalanobis distance function therefore adjusts for both scale of measurement of the variables and covariation among the variables. Use of this metric is equivalent to computing distances on variables transformed to their principal components.

The natural metric to use with the trace W criterion is the Euclidean distance. However, when using criteria based on multivariate statistical considerations, Mahalanobis is the natural metric to use.

When the clusters are distributed as p-variate normal and have equal covariance matrices, then Fisher's linear discriminant function is applicable, as is the Mahalanobis distance. The accuracy of the Mahalanobis metric is sensitive to the homogeniety of the cluster dispersions and decreases as the difference between the group dispersions increases. Recall the density function for the multivariate normal distribution

$$f(x;\mu,\Sigma) = \frac{1}{(2\pi)^{p/2}(\det \Sigma)^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

where Σ is the covariance matrix and μ is the mean vector of the distribution. Note the exponent which implies utilization of Mahalanobis distance is equivalent to

measurement of the density at the point x. The empirical distributions of the clusters will therefore determine the cluster to which the observation should be assigned. The following is a proof of the invariance of Mahalanobis distance under any non-singular linear transformation. Consider the transformation

$$Y = BX$$

and let $D(Y_i, Y_j)$ represent Mahalanobis distance between Y_i and Y_j.

$$\begin{aligned}
D(Y_i, Y_j) &= (Y_i - Y_j)^T C_Y^{-1} (Y_i - Y_j) \\
&= (BX_i - BX_j)^T C_Y^{-1} (BX_i - BX_j) \\
&= (X_i - X_j)^T B^T C_Y^{-1} B (X_i - X_j) \\
&= (X_i - X_j)^T B^T (BC_X B^T)^{-1} B(X_i - X_j) \\
&= (X_i - X_j)^T C_X^{-1} (X_i - X_j) \\
&= D(X_i, X_j)
\end{aligned}$$

Some other common metrics are defined below.

1. L_1 norm (city block)

$$D(X_i, X_j) = \sum_{k=1}^{p} |X_{ki} - X_{kj}|$$

2. L_p norm (Minkowski metrics)

$$D(X_i, X_j) = \left(\sum_{k=1}^{p} |X_{ki} - X_{kj}|^p \right)^{1/p}$$

3. Uniform norm

$$D(X_i, X_j) = \underset{k=1,2,\ldots,p}{\text{supremum}} \{|X_{ki} - X_{kj}|\}$$

III. THE DATA SET

A. ORIGIN

The present Student Opinion Form (SOF) system was started in the summer quarter of 1975 when it replaced the Student Instruction Report (SIR) obtained from the Educational Testing Service at Princeton. The SOF form has 16 questions and space for free-form comments from the students. The information obtained from the SOF data is used for the twofold purpose mentioned in Section I.A of this paper.

A SOF form (figure 2) should be completed by each student for each course segment he takes for credit. The term "course segment" is used because the same course may be offered to more than one group of students. To differentiate between the classes, segment numbers are assigned and a separate SOF identification number exists for each segment. Different segments of the same course may or may not be taught by the same professor. About 20 percent of the forms are not returned to administration officials due to lack of interest on the part of some students and instructors. Students have been informed that the results of the SOF data are used to assist in identifying faculty members for pay raises and tenure considerations.

Difficulties with legibility of the completed forms and with the OpScan machine have persisted for several quarters. The data available for this research has been coded with

STUDENT OPINION FORM
12ND NPS 5040/2(5-75)

INSTRUCTOR NAME ______________ COURSE No. ______________ SEGMENT ______________

PLEASE USE SOFT BLACK PENCIL

		NO COMMENT	STRONGLY AGREE	AGREE	NO STRONG OPINION	DISAGREE	STRONGLY DISAGREE
1.	The course was well organized	0	5	4	3	2	1
2.	Time in class was spent effectively	0	5	4	3	2	1
3.	The instructor seemed to know when students didn't understand the material	0	5	4	3	2	1
4.	Difficult concepts were made understandable	0	5	4	3	2	1
5.	I had confidence in the instructor's knowledge of the subject	0	5	4	3	2	1
6.	I felt free to ask questions	0	5	4	3	2	1
7.	The instructor was prepared for class	0	5	4	3	2	1
8.	The instructor's objectives for the course have been made clear	0	5	4	3	2	1
9.	The instructor made this course a worthwhile learning experience	0	5	4	3	2	1
10.	The instructor stimulated my interest in the subject area	0	5	4	3	2	1
11.	The instructor cared about student progress and did his share in helping us to learn	0	5	4	3	2	1

PLEASE USE THE FOLLOWING SCALE FOR THE NEXT FIVE ITEMS:

0. Not Applicable / Don't know / There were none		3. About Average	(Middle 40%)
5. Out standing	(Among the top 10%)	2. Fair	(In the lowest 30%)
4. Excellent	(Among the top 30%)	1. Poor	(In the lowest 10%)

		NA	O	E	A	F	P
12.	Overall, I would rate this instructor	0	5	4	3	2	1
13.	Overall, I would rate this course	0	5	4	3	2	1
14.	Overall, I would rate the textbook(s)	0	5	4	3	2	1
15.	Overall, I would rate the quality of the exams	0	5	4	3	2	1
16.	Overall, I would rate the laboratories	0	5	4	3	2	1

USE SPACE BELOW AND ON REVERSE SIDE FOR FREE FORM COMMENTS. IDENTIFY BY QUESTION NUMBER WHEN APPROPRIATE. These free form comments will be available only to the instructor.

THESE FOUR SPACES ARE AVAILABLE IF THE INSTRUCTOR WISHES TO SUPPLY ADDITIONAL QUESTIONS

17.	0	5	4	3	2	1
18.	0	5	4	3	2	1
19.	0	5	4	3	2	1
20.	0	5	4	3	2	1

PLEASE COMPLETE THE FOLLOWING ITEMS:

- THIS COURSE IS REQUIRED FOR ME
- THIS COURSE IS ELECTIVE FOR ME

DO NOT WRITE IN SHADED COLUMNS

QTR'S COMPT'D		HR'S THIS QTR		CURRIC			REPORT NO.					
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2
	3		3	3	3	3	3	3	3	3	3	3
	4		4	4	4	4	4	4	4	4	4	4
	5		5	5	5	5	5	5	5	5	5	5
	6		6	6	6	6	6	6	6	6	6	6
	7		7	7	7	7	7	7	7	7	7	7
	8		8	8	8	8	8	8	8	8	8	8
	9		9	9	9	9	9	9	9	9	9	9

Figure 2

indications where invalid responses occur. Only the valid information was considered in this thesis. Mean scores were computed for every instructor (every course segment) from the valid responses in each of the first 13 SOF items. Only the first 13 questions were used because of the high percentage of unusable responses in items 14, 15, and 16. Each of the responses recorded is an integer from one to five, with five being the upper (more desirable) end of the scale. These data are therefore considered on an ordinal scale. Table one categorizes the blocks of data which were available for this study. Note the short 3-digit notation to be used in this paper, indicating calendar year and quarter number.

CALENDAR YEAR	NUMBER OF RESPONDENTS	3-DIGIT CODE
Summer 1977	2440	773
Fall 1977	2967	774
Winter 1978	3056	781
Spring 1978	2964	782

Table 1

The majority of the analysis was performed using only quarter 773. Unless otherwise indicated, future references to the data set shall imply quarter 773 data.

B. TRANSFORMATION

The need for a common covariance structure when using the Mahalanobis metric has been emphasized. The transformation of quarter 773 data (which attempted to accomplish homogeneity of dispersions) is explained in this section.

The SOF data are 13-dimensional, and the best transformations would involve separate examination of each of the 13 variables. Due to the overwhelming complexity of this task, only a single transformation was sought.

In the SOF data the variance is very much a function of the mean. In fact, a course with a 5.0 mean vector has no variance whatsoever. Similar effects occur on the lower end of the scale. A variance-stabilizing transformation was sought which would help to relieve the dependence of the variance on the mean. Recall the normal distribution has independent mean and variance. Other well known distributions such as the Exponential, Geometric, and Poisson all have related mean and variance. The assumption of multivariate normality underlies much of standard classical multivariate statistical methodology. The effects of departure from normality are not clearly understood. Although marginal normality does not imply joint mormality, the presence of many types of non-normality is often reflected in the marginal distributions as well. The marginal distributions of the SOF data do not indicate any strong departures from normality.

Previous research by Professor R.R. Read [7] encountered the same need for a transformation of the SOF data. The

following transformation is due to Professor Read's findings:

$$\ln\left(\frac{x-1+a}{5+a-x}\right) \qquad \text{where} \quad a = .2 \tag{1}$$

The transformation was used on SOF item 12, and Bartlett's test substantiated the presence of homogeneity of variances. The groups involved here were the course segments, and the application was univariate.

Studies by Professor Glen Lindsay [8] and students in his course on Scaling Techniques produced results which suggested slight modifications to Professor Read's transformation.

$$\ln\left(\frac{x-1+a}{5+b-x}\right) \qquad \text{where} \quad \begin{aligned} a &= 2.0 \\ b &= 0.3 \end{aligned} \tag{2}$$

The same study could be described equally well with a constant second difference model, or what is the same thing, the function

$$x^2 + C \tag{3}$$

The three transformations were considered in the following manner. It was felt that the transformation which would produce the most nearly homogeneous covariance structure would be best. The three functions were applied to quarter

773 data, and then statistical tests for common covariance were administered. The test statistic comes from reference 5 and is explained in Appendix A. The results indicated that of the three, the first log transformation (1) generated the most nearly common covariance structure. The group structure whose covariance matrices were compared came from clusters formed by the MICKA algorithm (to be discussed in the next chapter). On the basis of the test results, the data were transformed by function (1), and all subsequent references to the data shall imply the transformed data.

Functions (1) and (2) are shown together on the graph in figure 3. The one chosen for use is the lower curve.

C. PRINCIPAL COMPONENTS

Recall the breakdown of the cross products matrix into the sum of the within and between scatter matrices. When considering the observations as individual SOFs (and the groups as courses), the cross products matrix will be called the Master scatter matrix with decomposition:

$$M = S + T$$

where S is the within course scatter and T is the between course scatter. It is reemphasized that, in this equation, the groups are the courses. The breakdown of the master scatter matrix may be examined before any clustering of course means is performed because the group structure

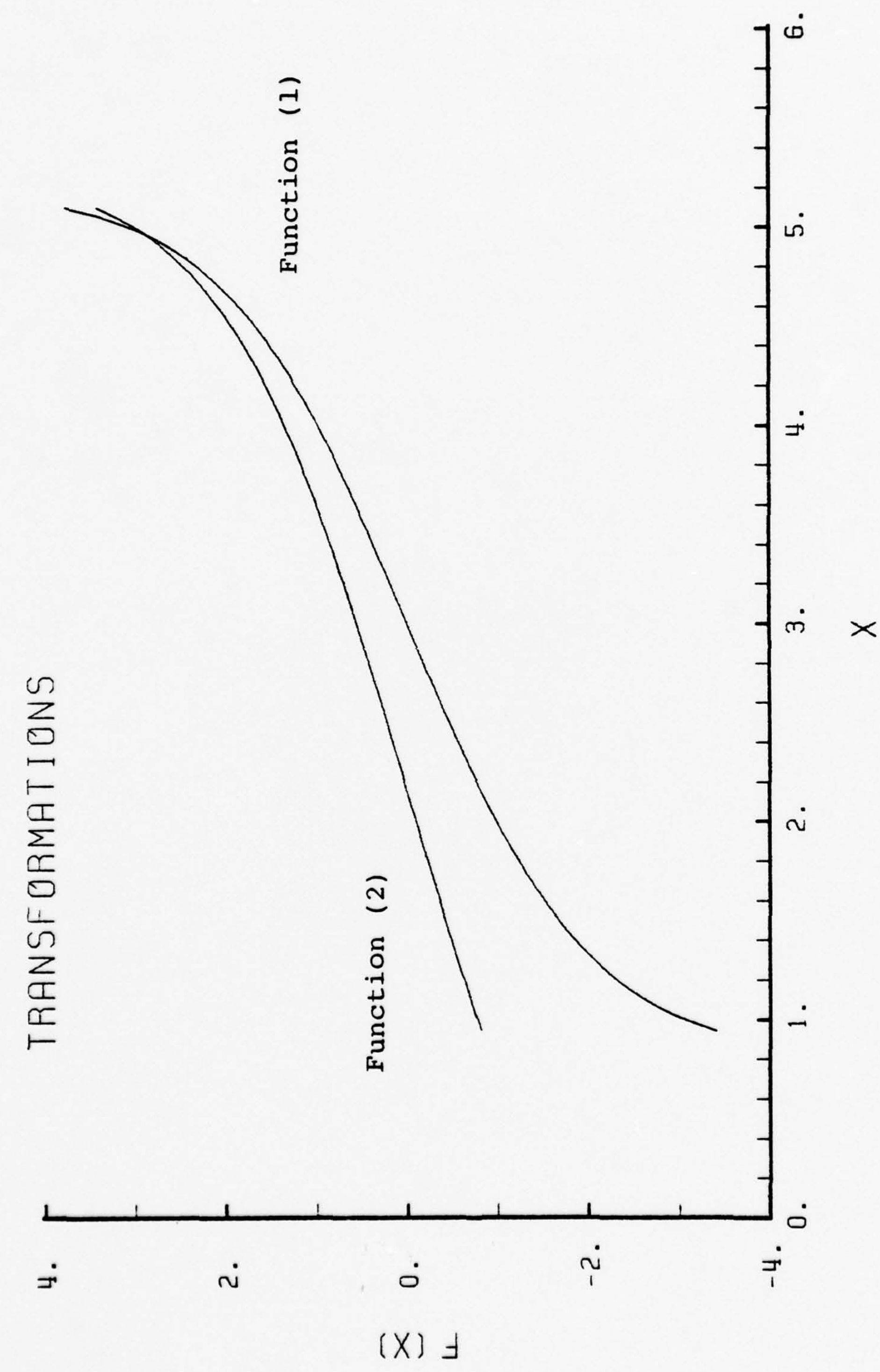


Figure 3

(courses are groups) is known. The discussion is enhanced by an algebraic description of the matrices involved. Let

$$M = \sum_{i=1}^{N} \sum_{j=1}^{ns_i} (X_{ij} - X_{..})(X_{ij} - X_{..})'$$

$$S = \sum_{i=1}^{N} \sum_{j=1}^{ns_i} (X_{ij} - X_{i.})(X_{ij} - X_{i.})'$$

$$T = \sum_{i=1}^{N} ns_i (X_{i.} - X_{..})(X_{i.} - X_{..})'$$

where

X_{ij} is the j-th SOF response form from the i-th course.

$X_{i.}$ is the mean vector of the i-th course.

$X_{..}$ is the grand mean.

ns_i is the number of students in the i-th course.

N is the total number of courses.

Since T represents the dispersion of the course means, it is the main object of the clustering efforts. It is natural to ask also, how much information is in S. To this end a principal components analysis was performed on the covariance matrices:

$$C_T = \frac{1}{N-1} T \qquad N-1 = 189$$

$$C_S = \frac{1}{\sum_{i=1}^{N} (ns_i - 1)} S \qquad \sum (ns_i - 1) = 1993$$

for quarter 773 data

Anderson [6] describes principal components as the axes of a coordinate system with special statistical properties. The principal components form a new coordinate system resulting from linear transformations of the variables which produce the special properties in terms of variances. The idea is to describe the data swarm by a new set of orthogonal coordinates so that the sample variances with respect to the new coordinates are in decreasing order. If the eigenvalues of the covariance matrix are ordered, i.e., $\lambda_1 > \lambda_2 > \ldots > \lambda_p$, then the variance in the new coordinate system is greatest in the dimension associated with λ_1, next greatest in the dimension associated with λ_2, etc. The sum of the eigenvalues is the total variance in the original coordinate system.

The results of the principal components analysis are shown in Table 2. First, it is of interest to compute how much of the total energy in M is accounted for by T.

$$\text{TOTAL} = 18.8(1993) + 156(189) = 66952$$

PRINCIPAL COMPONENTS ANALYSIS

	C_T EIGENVALUES	C_T EIGENVECTOR FOR λ_2	C_S EIGENVALUES	C_S EIGENVECTOR For λ_{13}
1	0.44	0.28	0.35	-0.27
2	136.97	0.30	0.47	-0.30
3	4.64	0.28	0.49	-0.28
4	0.46	0.29	0.51	-0.29
5	3.66	0.23	0.52	-0.19
6	2.57	0.19	0.63	-0.19
7	1.14	0.25	0.63	-0.24
8	1.63	0.27	0.68	-0.29
9	1.47	0.32	0.77	-0.33
10	0.66	0.30	0.89	-0.33
11	1.23	0.26	1.00	-0.28
12	0.96	0.35	1.10	-0.30
13	0.84	0.26	11.00	-0.27
TOTAL	156		18.8	

TABLE 2

T accounts for $29484 \div 66952 = 44$ percent of the total. This indicates that a great deal of variability must therefore be accounted for within the courses (i.e., with the students).

The principal components analysis of C_S shows the first principal component accounts for 55 percent of its total variance, but all other coordinate directions each account for 6 percent or less. Moreover, the direction of the first

component is essentially the main diagonal of 13 space, i.e., the signs are all the same and so are the magnitudes (approximately). Thus the data swarm may be thought of as an elongated ellipsoid directed along the main diagonal and having spheroidal (more or less) cross section. In particular, this suggests that the students within a course tend to score all 13 components more or less the same (all high, all moderate, or all low), but perceptions from student to student differ.

Turning to the principal components analysis of C_T, it is seen that 85 percent of the total variability is accounted for by the first principal component, and the second accounts for only three percent. Thus the data swarm of course means may be viewed as essentially one dimensional. Reference to its eigenvector reveals no single SOF item or group of SOF items is heavily weighted relative to the others and that the signs are again all the same. Thus, this component is similarly shaped along the main diagonal of 13 space, but more extremely elongated.

Some exploratory work was done on the within class variability (S) to see if the "number of quarters completed" by students has any effect on the variability represented by S. Figure four presents the results with a graph plotting within course variance versus time on board. Note the tendency for the variability to drop off in later quarters, possibly indicating more perfunctory completion of the forms.

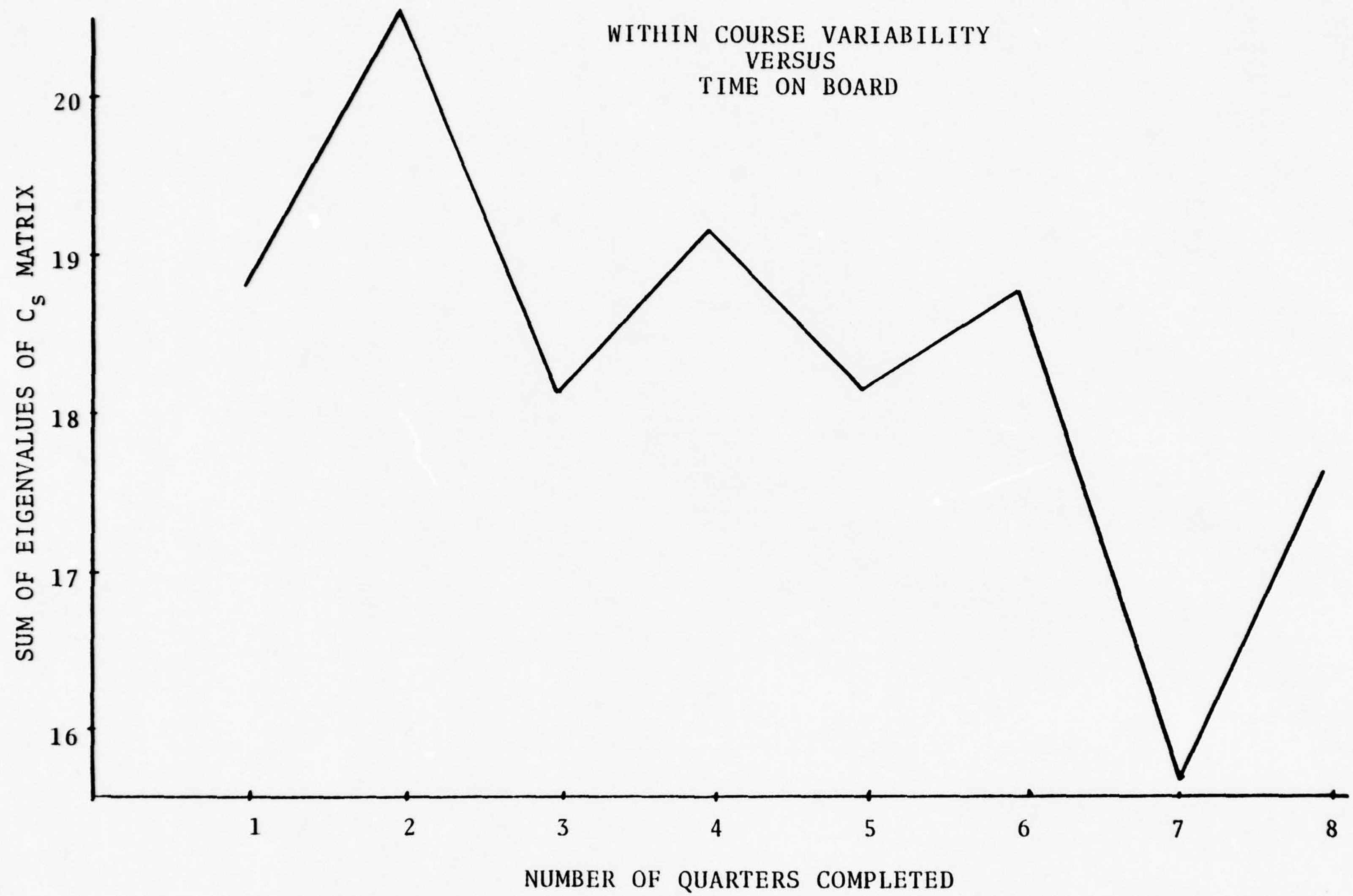

WITHIN COURSE VARIABILITY
VERSUS
TIME ON BOARD
SUM OF EIGENVALUES OF C_s MATRIX
20
19
18
17
16
1
2
3
4
5
6
7
8
NUMBER OF QUARTERS COMPLETED

FIGURE 4

IV. THE MIKCA METHOD

A. THE ALGORITHM

The specific algorithm chosen for the cluster analysis is the MIKCA (Multivariate Iterative K-MEANS Clustering Algorithm) program written by Douglas J. McRae as a part of his doctoral dissertation at the University of North Carolina, Chapel Hill.

Reference to the flow chart in figure 5 will aid the reader in the following discussion of the algorithm. Inputs to the program are the data matrix, an estimate for g (the number of clusters), and choice of criterion and distance functions.

In the first step, preliminary calculations are made, such as the variable means and standard deviations, as well as the cross products matrix T. The next step forms the initial cluster solution. A random choice of s observations serves as the initial cluster centers. Then each of the other observations is assigned to the nearest cluster. Euclidean distance is used for this initial phase, and the cluster centroids are recomputed after each observation is assigned to a group. The observations are considered in the same order as they were input. After all of them have been assigned to clusters, the criterion value is computed. This initial cluster-finding technique is referred to as a one-pass K-MEANS procedure. It is performed three times, and

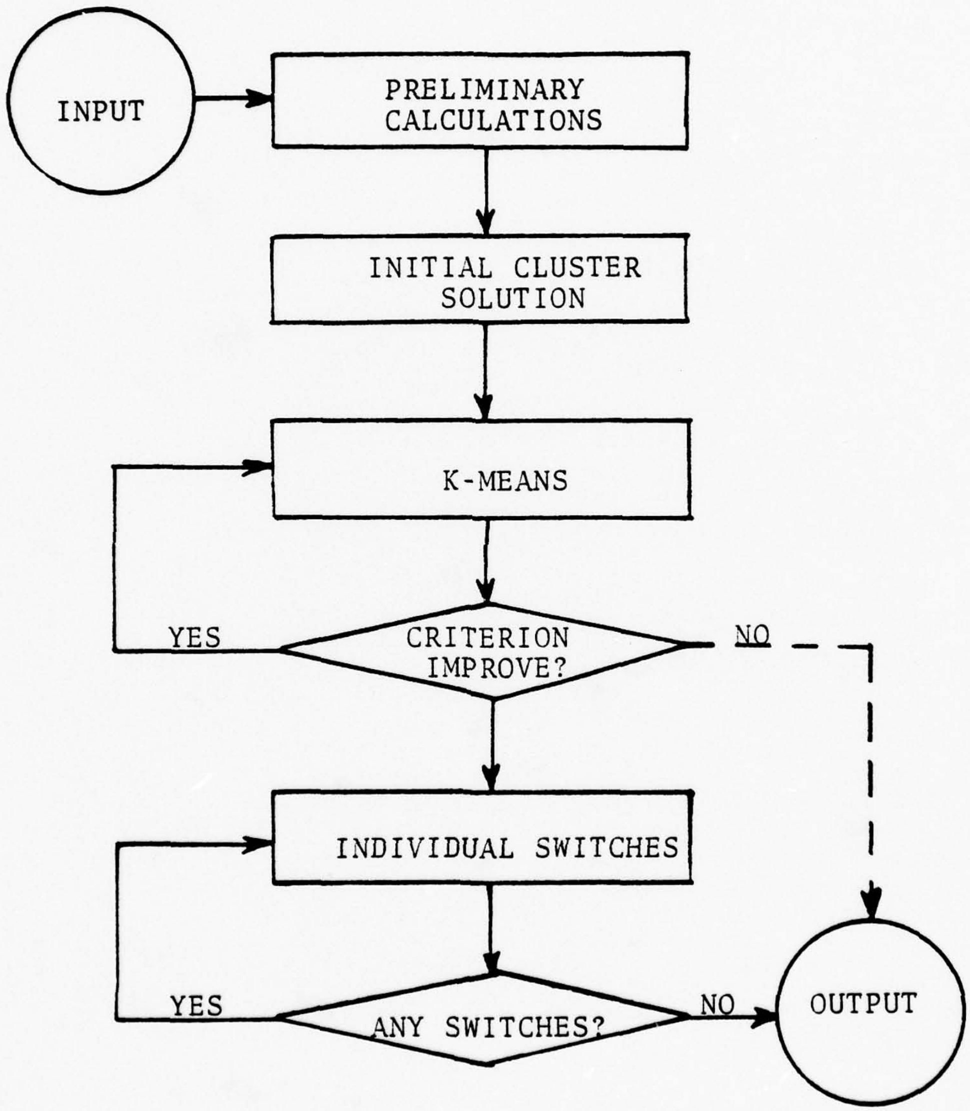


FIGURE 5 MIKCA FLOW CHART

the solution which yields the best criterion value is chosen as the initial cluster solution.

After the initial solution has been found, the program advances to the iterative K-MEANS phase where the observations are again considered in the order in which they were input to the program. It is this phase where the user's choice of distance function is used. The distance from each observation to each cluster centroid is again computed, this time with the user's distance function, the assignment to the closest centroid being made and the centroid updated to reflect its new membership. After considering all n observations in this manner, the new criterion value is checked for possible improvement during the K-MEANS iteration. As long as the criterion value improves, the K-MEANS procedure is repeated; if the criterion fails to improve then the MIKCA algorithm goes to the next step, the individual switches section.

Note the importance of the order of consideration of the observations. The order is important because the cluster means are recomputed after each observation is reassigned.

In the individual switches phase, consideration is given to moving each observation to every other cluster, the move being made if and only if an improvement in the value of the criterion results. An elaborate labelling procedure provides a unique order in which to consider each observation. This procedure continues until a complete pass through the data is made with no changes in cluster membership.

The MIKCA algorithm provides the following options for distance and criterion functions.

CRITERION

1. Minimum trace W
2. Minimum det W
3. Maximum largest order of $|B-\lambda W| = 0$
4. Maximum sum of roots of $|B-\lambda W| = 0$

DISTANCE

1. Euclidean
2. Weighted Euclidean
3. Mahalanobis

Using R.A. Fisher's iris data, McRae tested his algorithm and produced extremely good results. Using the det W criterion and Mahalanobis distance, MIKCA produced a solution identical to the classification given by multiple discriminant analysis. This is a notable achievement since the cluster procedure, which does not know the true composition before the analysis, makes the same final classification of observations as does the discriminant procedure, which bases its analysis on the group composition information.

The MIKCA provides as output the value of the criterion function, the cluster membership, and the cluster mean vectors. Also provided are two matrices, T and W. The program was written in FORTRAN IV for the IBM 360 series of computers.

B. MODIFIED MIKCA

Initially, the MIKCA program was used with the p-component mean responses for each course as the input data matrix. Since the number of students utilized in producing these means is quite variable, these input vectors are not equally well determined and, as has been mentioned earlier, may effect the covariance structure between the objects. It is desirable to have the option of weighting these course means in order to effect a better balance in terms of their accuracy and to reduce any consequential distortion in the covariances. It is convenient to refer to this modification as the "1 man 1 vote" option, and to the original technique as the "1 course 1 vote" option. The following algebraic definitions will aid in illustrating the weighting effect.

Recall the breakdown of the master scatter matrix into the sum of within and between matrices.

$$M = S + T$$

When the mean scores are computed for each course and used as inputs to MIKCA, then a different dispersion structure takes form. The groups are no longer the known courses, but are now the object of the problem. The groups are unknown clusters of courses (or professors). Let T^* denote the total scatter contained in the data when each observation represents a course mean vector. T^* may also be expressed as the sum of within and between scatter matrices.

$$T^* = W^* + B^*$$

These matrices are defined as follows:

$$T^* = \sum_{s=1}^{g} \sum_{k=1}^{nc_s} (\bar{x}_{sk} - \bar{x}_{..})(\bar{x}_{sk} - \bar{x}_{..})'$$

$$W^* = \sum_{s=1}^{g} \sum_{k=1}^{nc_s} (\bar{x}_{sk} - \bar{x}_{s.})(\bar{x}_{sk} - \bar{x}_{s.})'$$

$$B^* = \sum_{s=1}^{g} nc_s(\bar{x}_{s.} - \bar{x}_{..})(\bar{x}_{s.} - \bar{x}_{..})'$$

where

nc_s is the number of observations (courses) in the s-th cluster.

g is the number of clusters.

$\bar{x}_{sk}$ is the k-th observation (course mean vector) in the s-th cluster

$\bar{x}_{s.}$ is the mean vector of the s-th cluster

$$\left(\frac{\sum_k \bar{x}_{sk}}{nc_s}\right)$$

$\bar{x}_{..}$ is the grand mean

$$\left(\frac{\sum_s \sum_k \bar{x}_{sk}}{\sum_s nc_s}\right)$$

Note that the grand mean mentioned here is not the same as the grand mean used in the decomposition of the master matrix M. The difference between T and T* is that T is weighted by the number of students in each course, ns_i. This weighting factor was lost when the individual observations were viewed as the class mean vectors. A close algebraic examination of T will illustrate its weighted property. Originally, we had M = S + T where:

$$T = \sum_{i=1}^{N} ns_i (x_{i.} - x_{..})(x_{i.} - x_{..})'$$

It is now helpful to show the decomposition of T.

$$T = W + B$$

Let $x_{i.}$ become $\bar{x}_{sk}$ (k-th course mean in s-th cluster) and ns_i become ns_{sk} (number of students in k-th course of s-th cluster). Therefore the same T can be reexpressed as

$$T = \sum_{s=1}^{g} \sum_{k=1}^{nc_s} ns_{sk} (\bar{x}_{sk} - x_{..})(\bar{x}_{sk} - x_{..})'$$

Letting

$$W_s = \sum_{k=1}^{nc_s} ns_{sk} (\bar{x}_{sk} - \bar{x}_s)(\bar{x}_{sk} - \bar{x}_s)'$$

where

$$\bar{x}_s = \frac{\sum_{k=1}^{nc_s} ns_{sk}\,\bar{x}_{sk}}{\sum_{k=1}^{nc_s} ns_{sk}} \quad \text{(weighted mean vector of s-th cluster)}$$

then

$$W = \sum_{s=1}^{g} W_s$$

and

$$T = W + B \qquad \text{(B is obtained by subtraction)}$$

The understanding of this distinction is important because it describes the abbreviated (unweighted) dispersion upon which MIKCA bases its cluster solution.

A number of changes were made to the MIKCA computer code to allow for a system of weights, ns_i, for the course means. The modified code extends the capability of MIKCA by making this option available. It amounts to using T rather than T* as the basic dispersion structure. This seems more natural because the matrix T appears in the earlier decomposition.

$$M = S + T = S + W + B$$

Some of the changes are summarized here:

1. Allow for class size as input.
2. Alter the computation of T to allow for the weighting factor.
3. Alter the computation of cluster centroids to allow for weighting.
4. Alter calculations of the B matrix for the same reason. (W is found by subtracting B from T.)

The computer code for the modified MIKCA is included in Appendix F.

Cluster solutions using both weighted (T) and unweighted (T*) dispersion structures were found and compared (see table 3 in next section). The comparison indicates some differences in cluster solutions, however the importance of these differences is left to the reader.

C. RESULTS

Several cluster solutions were formed using the MIKCA algorithm. It seemed wise to include the number of students in a course as the 14-th variable. The natural logarithm of the class size was the transformation applied to this variable. Since class sizes ranged from two to 40, this transformation brought the values into a similar range as the other 13 variables and also reduced skewness. For quarter 773 the mean class size was 12.7 students with a standard deviation of 7.9. For the transformed variable these

statistics are 2.3 and 0.7. Cluster solutions were found with and without inclusion of this 14-th variable. The results are shown in table three.

Another option available to the MIKCA user is the standardization of variables prior to entering the clustering process. McRae points out how this option becomes very useful when the variables are on vastly different scales of measurement. Except for the 14-th variable the present scales are psychological in nature and seem to be much the same. Some exploratory work was performed with the standardization option (see table three) but it was not considered significant because of the similarity in the scales of measurement.

Table three shows the comparisons of cluster results obtained under various conditions. The comparison coefficient provides a measure of agreement between solutions and is computed by a method introduced in Chapter VII. Table three shows generally higher values for g = 3, indicating that there exists robustness of solutions for the smaller values of g.

The results of these cluster solutions may also be seen in graphical form by referring to Appendix B. These graphs, called profile charts, depict the mean vectors for each of the clusters formed by the MIKCA algorithm. The mean vectors have been standardized so that one can see the number of standard deviations from the grand mean. These profiles are

COMPARISON COEFFICIENTS FOR CLUSTER SOLUTIONS
OBTAINED UNDER VARIOUS CONDITIONS

	13N	13S	14N	14S
13N	1.0	1.0	.75	.83
13S		1.0	.75	.83
14N			1.0	.86
14S				1.0

	13N	13S	14N	14S
13N	1.0	1.0	.63	.61
13S		1.0	.63	.61
14N			1.0	.58
14S				1.0

	13N	13S	14N	14S
13N	1.0	.57	.72	.82
13S		1.0	.44	.48
14N			1.0	.75
14S				1.0

COMPARISON OF
ORIGINAL MIKCA
AND
MODIFIED MIKCA
SOLUTIONS

COMPARISON COEFFICIENT=.57

LEGEND

13N 13 VARIABLES, NOT STANDARDIZED
13S 13 VARIABLES, STANDARDIZED
14N 14 VARIABLES, NOT STANDARDIZED
14S 14 VARIABLES, STANDARDIZED

TABLE 3

also helpful in identifying the variables which are significant in the cluster membership. For example, an important variable would be one that produces a break in the pattern.

In the 13 variable case, the profiles produced results which indicated the lack of clearly dominant variables in cluster identification. With introduction of the 14-th variable, some very revealing results become immediately apparent. While the cluster membership changed little in going from 13 to 14 variables, the cluster with the highest mean vector became clearly associated with the smallest class sizes. Similarly the cluster with the lowest mean vector is characterized by a very large class size. This finding is one of the most significant results.

One of the most critical decisions facing the analyst is the number of clusters to form. Some algorithms based on the K-MEANS idea allow g to change during the clustering process, however the MIKCA method requires g to be input by the user and it does not change in the course of the program execution. Typically the investigator does not know the number of clusters in the data, and he must make some educated guess. As pointed out earlier, it is possible for several different, but meaningful, cluster solutions to exist in one body of data.

The method used to determine g was to obtain solutions based upon several values for g and then plot the criterion values for each of these solutions. An appropriate choice for g would be a number beyond which the marginal improvement

of the criterion becomes insignificant. Figures 6 and 7 are the results of such tests suggesting that six clusters represent the major portion of the separating power of the algorithm.

Profile charts of the cluster solutions with g = 6 were uninteresting. The middle clusters were all bunched together suggesting that clusters were forced on that part of that data where perhaps they did not actually exist (i.e., sparse data near the boundaries). Comparison results (table 3) indicate a much more stable solution when g is reduced below six.

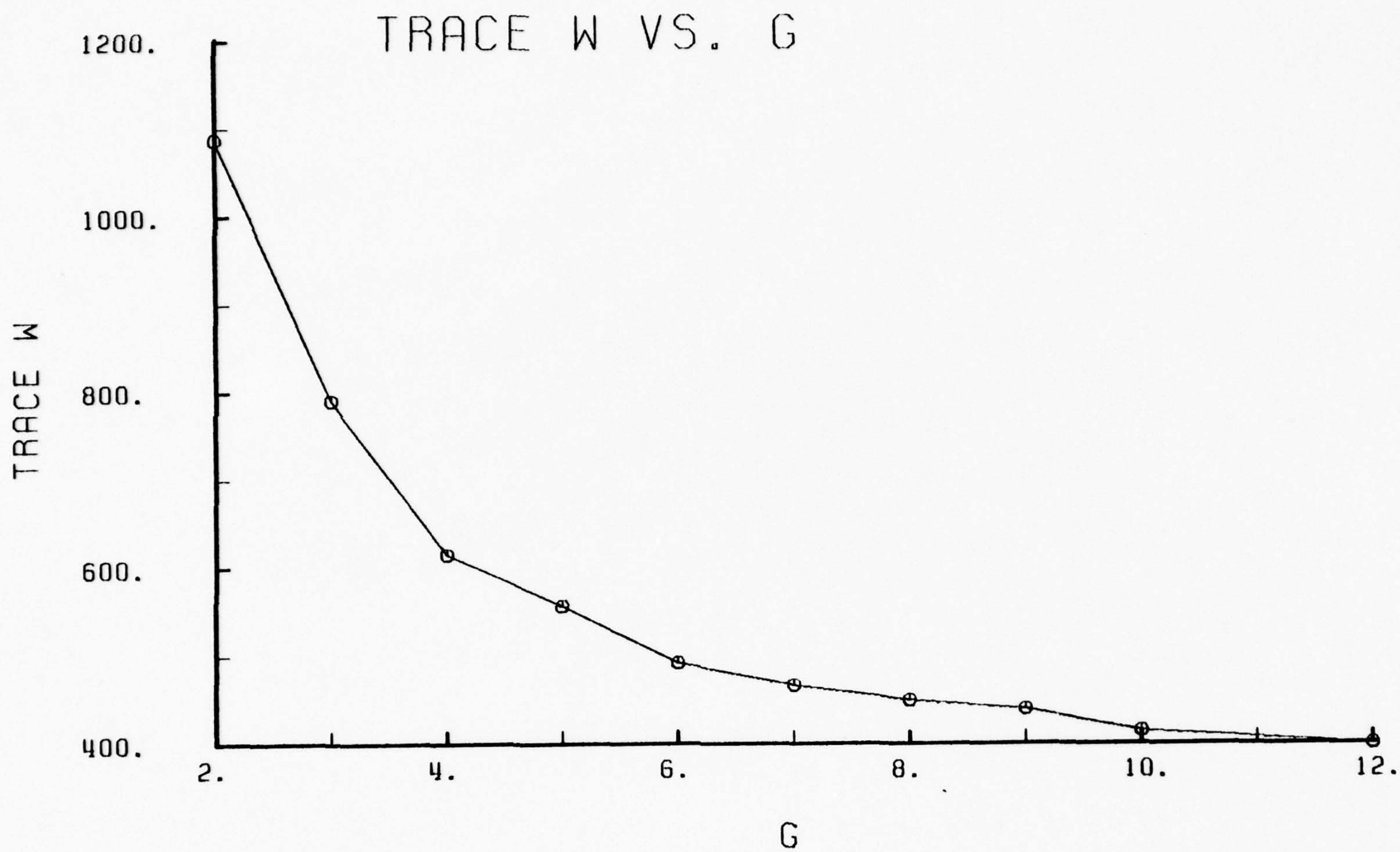


Figure 6

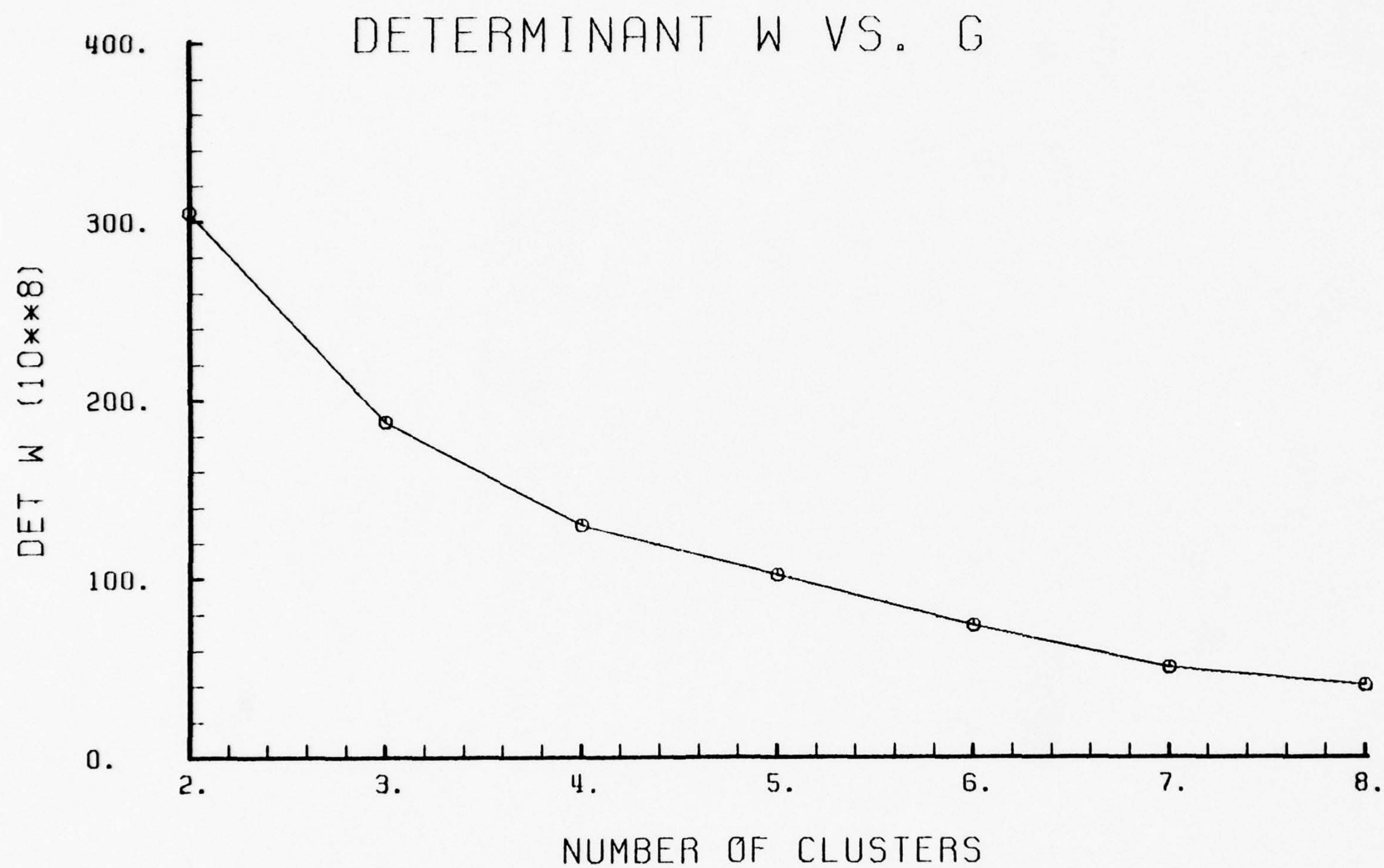

DETERMINANT W VS. G
400.
300.
200.
100.
0.
DET W (10**8)
2.
3.
4.
5.
6.
7.
8.
NUMBER OF CLUSTERS

Figure 7

V. DISCRIMINANT ANALYSIS

A. THEORY

As mentioned earlier, discriminant analysis allows an analyst to classify new observations based on observations which are samples from known groups. Only Fisher's linear discriminant function will be used in this study. It also provides information about the relative importance of the various variables in assigning an observation to a group. The linear discriminant function is based on the assumptions of multivariate normality and homogeneity of dispersions. The ability to identify the dominant variables and the dimension reduction offered by the discriminant space were both extremely useful aids for analyzing the SOF data.

These "more important" variables will be earmarked for later use in the construction of Chernoff's FACES. Also of interest is the plot of data points in discriminant space. The interaction of the coefficients in the discriminant functions will be seen as well as the characterization of the dimensions.

In order to describe our usage of discriminant analysis, let us first suppose there are only two clusters in 13-dimensional space. It is deisred to project these two clusters orthogonally onto a line so that the variation between the two groups is as large as possible relative to the variation within the two groups. Finding the direction of projection to accomplish this is part of the purpose of

discriminant analysis. The solution provides a way of discriminating between the two clusters by a suitable linear combination of the 13 variables. The same theory is generalized to g groups, where Wilks [9] has shown that a projection to the smaller of g-1 or p dimensions is possible without loss of information. Recall the earlier discussion that indicated this smaller number as t, the number of nonzero eigenvalues of $W^{-1}B$. The eigenvalues are the variances in the direction of their associated eigenvectors. One can easily determine the proportion of variance attributable to each of the dimensions of discriminant space and also the SOF items which load most heavily in each dimension.

One gains insight into the variables from examination of the coefficients in the discriminant functions. There is one function for each dimension, the standardized coefficients of which are used in this analysis.

B. RESULTS

Up to this point, most of the analysis has been performed on the 190 courses in quarter 773. A smaller, more manageable data base was needed to continue. Also, it seemed wise to prepare to study individual departments. The Electrical Engineering Department was chosen for further analysis since it is a large department and hence not too small for this purpose. Over the four quarter period, there were 116 course segments with valid SOF responses. These 116 courses from the EE department were the data used in the discriminant analysis.

When dealing with four clusters, the dimensionality of the discriminant space is three, and depending on the size of the eigenvalues, perhaps fewer dimensions will provide sufficient discrimination. Table four gives the results of performing a discriminant analysis. Figure eight is a graph of the two-dimensional discriminant space (the third dimension is neglected).

The eigenvalues indicate 94 percent of the total variance is represented by the first two discriminant functions. Figure eight corroborates this fact by depicting easily seen separation in two dimensions. Imagine projecting the points onto the horizontal axis. Discrimination in the first dimension would account for 73.6 percent of the variation. Groups one and four would easily be separated, but two and three would overlap.

Examination of the coefficients will enable one to label the dimensions by identifying the dominant characteristics which they measure. The first dimension is along the horizontal axis and is associated with the first discriminant function. The magnitude of the coefficients indicates their relative impact on the dimension. The signs aid in understanding which variables reinforce one another (matching signs) and which tend to cancel (opposite signs). In the first function of table four, SOF item 12 is the most prominent. This question (see figure 2) asks the student to score the overall rating of the instructor. It is not surprising

RESULTS OF DISCRIMINANT ANALYSIS ON THE 116 COURSES
IN EE DEPARTMENT OVER A FOUR QUARTER PERIOD

DISCRIMINANT FUNCTION	EIGENVALUE	RELATIVE PERCENTAGE
1	5.79	73.6
2	1.64	20.8
3	0.44	5.6

STANDARDIZED DISCRIMINANT
FUNCTION COEFFICIENTS

	Function 1	Function 2	Function 3
1	-0.11	0.23	-0.22
2	0.13	0.14	-0.09
3	-0.05	1.46	0.01
4	-0.47	-0.36	-0.80
5	-0.31	0.01	-0.82
6	0.08	0.36	-0.74
7	0.15	-0.36	0.12
8	0.23	1.15	-0.36
9	0.36	-0.82	-0.47
10	0.05	0.08	-0.77
11	-0.20	-1.16	1.18
12	-0.72	-1.08	1.80
13	-0.18	0.91	0.92

TABLE 4

TWO DIMENSIONAL DISCRIMINANT SPACE

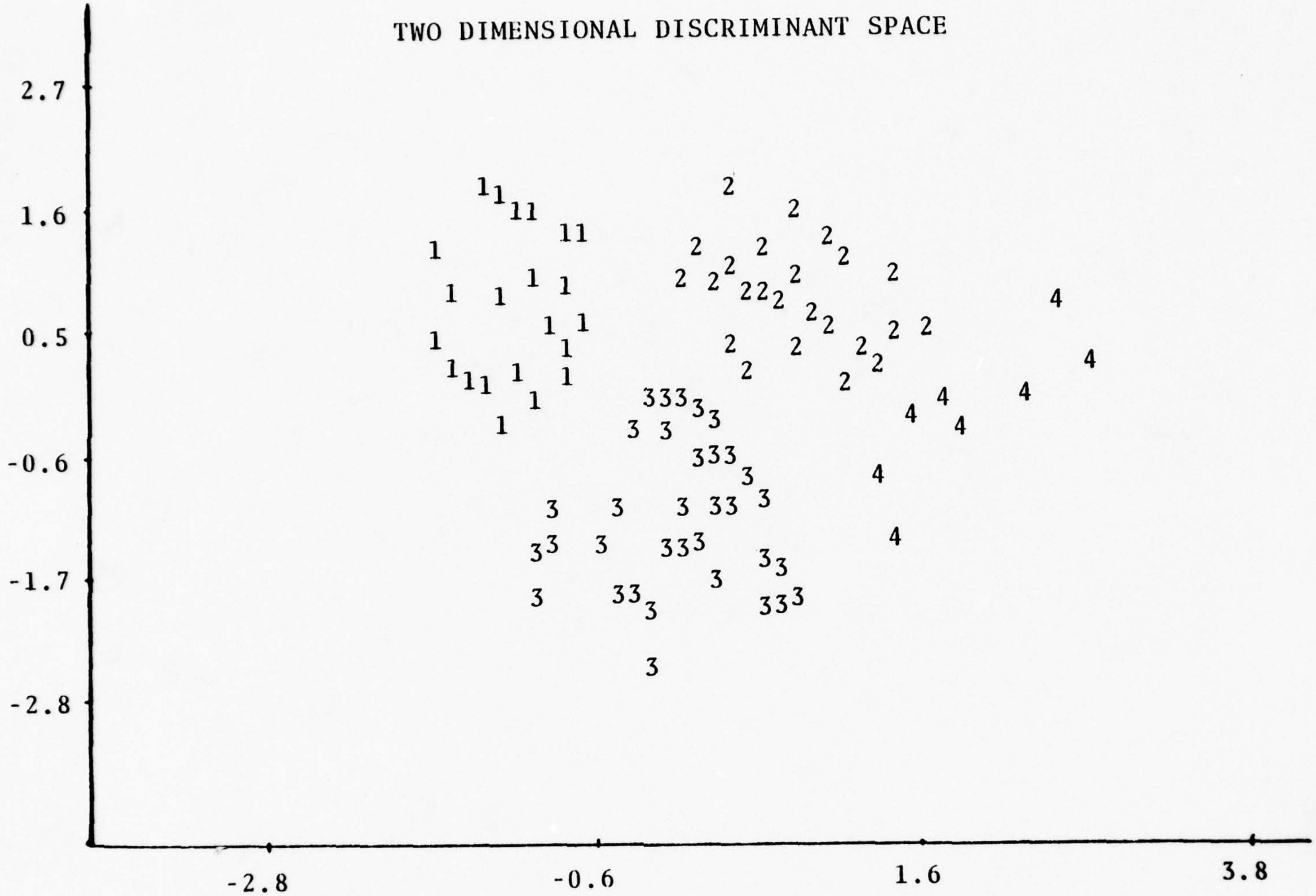


FIGURE 8

that this question is very important in the discrimination process. High marks on items four and five tend to reinforce a high mark on question 12. Those questions are:

(4) Difficult concepts were made understandable.

(5) I had confidence in the instructor's knowledge of the subject.

Interestingly, a high mark on item nine (instructor made the course a worthwhile learning experience) tends to diminish the effect of a high score on item 12. The first dimension is dominated by question 12 and was labeled the "popularity" dimension.

The second dimension is depicted by the vertical axis on the graph in figure eight, and measurements along this dimension are controlled by the second discriminant function. The separating power in this direction is less than one third that of the first. Note however that the vertical scale is compressed 25 percent more than the horizontal scale (1.5 inches vertical = 2.0 inches horizontal). Items three and eight has strong positive coefficients whereas questions 11 and 12 are pulling heavily in the negative direction. However the strength of the information is not great, and deeper interpretation hardly seems worth the effort.

Only 5.6 percent of the total variance appears in the third function, and it is therefore considered insignificant. One might note that item 12 also dominates the third dimension.

The main purpose here has been to identify variables for use in constructing Chernoff FACES. The discriminant analysis has served that purpose well, and it has also described the character of the dimensions.

VI. CHERNOFF FACES

A. BACKGROUND

Chernoff's FACES was the second cluster method to be applied to the SOF data. The method was used with the same purpose in mind, and it was hoped that earlier cluster solutions could be reproduced by this method. Additionally, there was the possibility of gaining new information about the structure within the data. Professor Herman Chernoff developed this graphical method for representing multivariate data. The now familiar data point in p-space is represented by a computer-drawn cartoon of a face whose characteristics (features) are determined by the position of the point. Features such as nose length and mouth curvature correspond to components of the data point. In the case of the SOF data, each component of the 13-dimensional vector can be made to control one of 20 features, and seven constants can be selected for the remaining features. The technique lends itself to clustering since the investigator can group together those faces which resemble each other.

Chernoff [10] points out that people spend a great deal of their life studying and reacting to faces. The human mind subconsciously acts as a high speed computer sometimes detecting barely measurable differences and ignoring unimportant differences, even if they are large. Chernoff claims that unlike a machine, the mind has the capability

to disregard non-informative data and search for useful information. He states that certain major characteristics of the faces are instantly observed and easily remembered; finer details and correlations become apparent after studying the faces. Clustering by sorting the faces is certainly easier than staring at a large matrix of data. The method has pitfalls and limitations and some of them will be dealt with in this thesis.

After the publication of Chernoff's method [11], quite a number of people began experimenting with the technique. Lake [12] mentions a few more successful applications of Chernoff's method, including:

1. L.A. Bruckner of Los Alamos Scientific Lab of the University of California while studying the performance of offshore oil companies.
2. Johns Hopkins University
    a. Developing methods of psychiatric screening.
    b. Monitoring patients in intensive care units.
    c. Monitoring the stock market.
3. Dr. David L. Huff of the University of Texas in developing urban regional indicators that measure the quality of life.
4. Professor P.C.C. Wang and Gerald Lake at the Naval Postgraduate School in analyzing Soviet naval penetrations into the Indian Ocean and the African littoral; and Soviet foreign policy in sub-Saharan African states.

5. Professor Chernoff in geological and fossile-related experiments.

The field of computer graphics has experienced tremendous growth in recent years due mainly to the state of the art in computers and computer display equipment (including both video and plotting types). The adage that "a picture is worth a thousand words" has proven to be quite true. Recent developments include on-line programs that perform statistical analysis with polygon, bar graphs, arrows, and scatter diagrams. Three-dimensional data displays have facilitated the work of engineers and statisticians alike.

An interesting application of the FACES program is Bruckner's study of offshore drilling by oil companies. Figure 9 displays some of his results. Two of the features, nose width and eye separation, are controlled by the variables "expected years to production" and "number of leases won", respectively. Other features are controlled by a variety of variables representing the company's financial health and growth potential.

Reference to figure 10 will help describe how the faces are constructed. Table 5 gives the range of the variables which control the features and distance parameters of the face.

The data are first converted to the X parameters as follows. The variable Z is used to control the parameter X_i which is allowed to range from a_i to b_i.

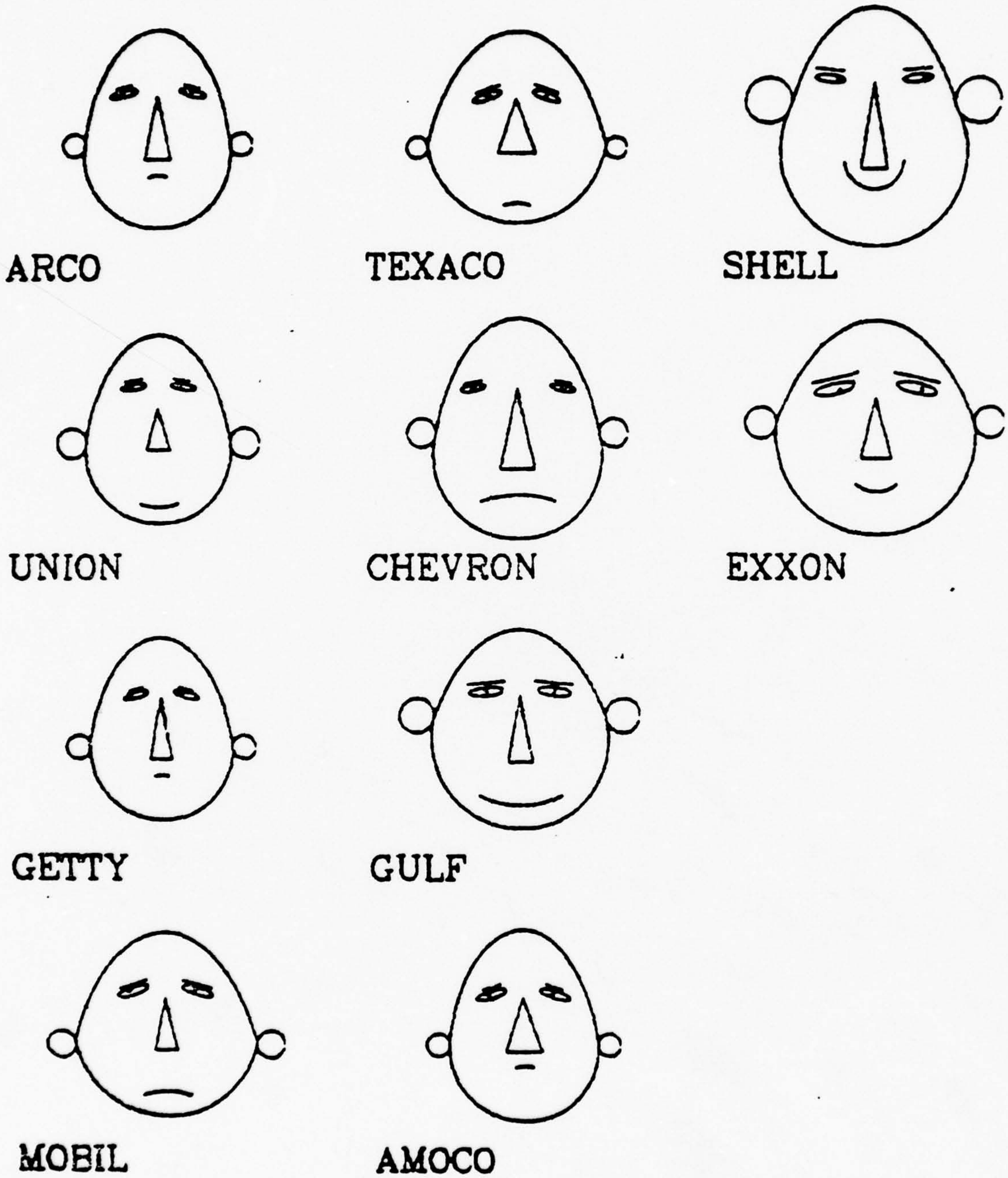


Figure 9

Bruckner's Offshore Hydrocarbon Producers

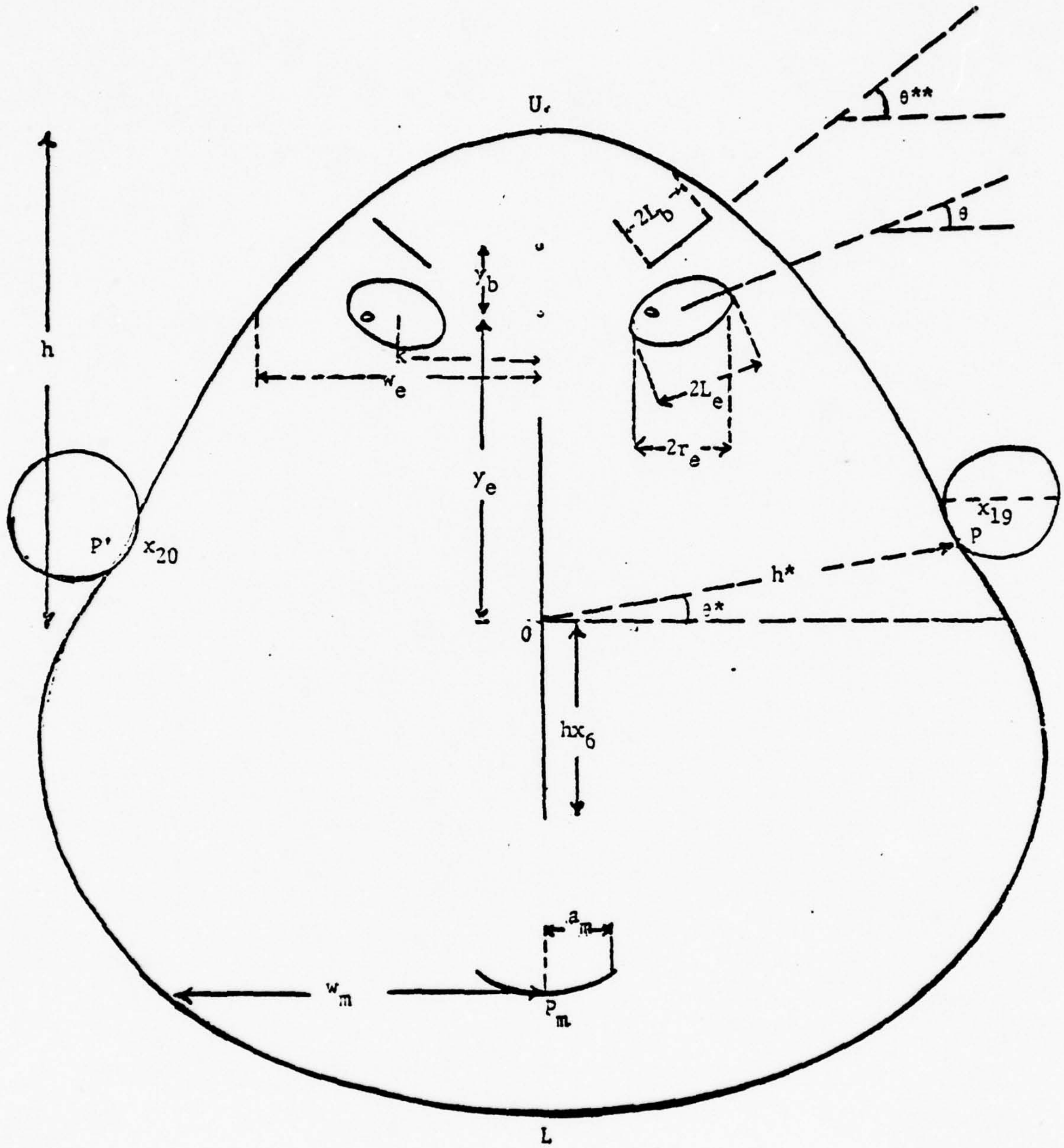


Figure 10

Chernoff Face with Ears

FEATURE RANGES AND DESCRIPTION

This table is taken from reference 10, and the descriptions are not complete. For a more detailed, mathematical explanation, see Appendix C.

Range			
(0,1)	x_i	controls h^*	distance from 0 to P
(0,1)	x_2	controls θ^*	angle between OP and horizontal
(0,1)	x_3	controls h	half-height of face
(0.5,2)	x_4	is	eccentricity of upper ellipse of face (width/height)
(0.5,2)	x_5	is	eccentricity of lower ellipse of face (width/height)
(0,1)	x_6	controls	length of nose
(0,1)	x_7	controls P_m	position of center of mouth
(-5,5)	x_8	controls	curvature of mouth (radius = h/x_8)
(0,1)	x_9	controls a_m	length of mouth
(0,1)	x_{10}	controls y_e	height of centers of eyes
(0,1)	x_{11}	controls x_e	separation of centers of eyes
(0,1)	x_{12}	controls θ	slant of eyes
(0.4,0.8)	x_{13}	is	eccentricity of eyes (height/width)
(0,1)	x_{14}	controls L_e	half-length of eye (L_e also depends in part on x_{10} and x_{11})
(0,1)	x_{15}	controls	position of pupils
(0,1)	x_{16}	controls y_b	height of eyebrow center relative to eye
(0,1)	x_{17}	controls $\theta^{**}-\theta$	angle of brow relative to eye
(0,1)	x_{18}	controls	length of brow
(0,1)	x_{19}	is	ear diameter
(0,1)	x_{20}	is	nose width

TABLE 5

$$X_i = a_i + (b_i - a_i)\left(\frac{Z - m}{M - m}\right)$$

where m and M are the observed minimum and maximum of Z.

Chernoff's technical report [10] presents a very detailed description of the geometric relationship of the features in the face construction. A few general remarks concerning the geometric attributes are included here. The boundary of the face is formed by joining portions of two ellipses, an upper and a lower. The angle theta (θ) determines where the ellipses meet and consequently, the height of the ears. The nose is a triangle centered at the origin. Both its height and width are variable. The curvature of the mouth is a portion of a circle, the radius and center of which are also variable. The eyes are formed by ellipses whose angle, half-length, and eccentricity are all controlled by variables.

B. FEATURE-VARIABLE RELATIONSHIP

A frequent question is whether some features are more informative than others. Some observers feel that the eyes convey the most information; others regard the mouth or the shape of the face as the most relevant feature. The results of the discriminant analysis identified the most dominant variables in the discriminant space. Now these variables must be assigned to facial features.

Chernoff [13] himself conducted an experiment to evaluate the effect on classification error of random permutations in the assignment of variables to features. He found that

random permutations would change the faces so that a classifier might increase or decrease his number of errors by a factor of about 25 percent. Unfortunately, his experiment did not evaluate the efficiency of specific features. His studies also make no effort to determine whether ability to discriminate depends on the dimensionality of the data.

Considering Chernoff's findings, it would seem that the assignemnt of variables to features is of minor importance. The use of discriminant analysis provides a way of detecting which variables are important, and it seems appropriate to take advantage of this valuable information when constructing the faces. Moreover, there is choice in the features that are selected for use. The author's choice of the six best features are starred in table 6. The table gives the complete list of feature-variable combinations. The results of the discriminant analysis were relied upon heavily in forming the variable assignments.

Reference to figure eight (discriminant space) and table 6 will aid in the following discussion. In the first dimension the important SOF items are 12 and 4 which control the mouth curvature and ear height, respectively. High scores on these two items separate the observation well to the negative end of the scale and cause the face to have a big smile and high ears. Items 12 and 4 have the same sign (negative) but item 9 is associated with a large positive coefficient and controls the lower eccentricity of

FEATURE-VARIABLE COMBINATIONS

	FEATURE	THREE DIFFERENT TRIALS CONTROLLED BY 13 VAR	6 VAR	8 VAR
1	FACE WIDTH	0.5	0.5	0.5
2	ANGLE θ	4	0.65	4
3	FACE HEIGHT	0.7	0.7	0.7
4	UPPER ECCENTRICITY	8	0.95	0.95
*5	LOWER ECCENTRICITY	9	4	0.6
6	NOSE LENGTH	10	0.45	9
7	MOUTH CENTER	0.5	0.3	0.5
*8	MOUTH CURVATURE	12	12	12
9	MOUTH LENGTH	13	0.7	0.8
10	EYE HEIGHT	0.23	0.23	0.23
11	EYE SEPARATION	1	0.5	0.5
*12	EYE SLANT	11	3	3
13	EYE ECCENTRICITY	3	0.6	0.6
14	EYE HALF LENGTH	6	0.5	5
*15	PUPIL POSITION	2	9	13
16	EYEBROW HEIGHT	0.3	0.3	0.3
*17	EYEBROW ANGLE	5	8	8
18	EYEBROW LENGTH	0.4	0.4	0.4
19	EAR DIAMETER	0.3	0.3	0.3
*20	NOSE WIDTH	7	11	11

Integer numbers are the SOF item #.
Decimal values are the fixed features

TABLE 6

the face. A low mark on this item would complement high marks on items 12 and 4 and would be reflected in the lower face having small eccentricity (more narrow).

Turning to the vertical axis (second dimension) of figure eight which has 21 percent of the total variance, the dominant variables are 3, 8, and 11, where 11 is negative; 3 and 8 are positive. High scores on items 3 and 8 separate the observation upward on the vertical axis and are reflected as highly eccentric eyes and upper ellipse.

Droopy eyes, reflecting a small value for SOF item 11, tend to complement and reinforce the higher values for items three and eight. It seems like a good idea to use the results of the discriminant analysis in this way, but it is impossible for the viewer to know which variables act together and which interfere unless he is told beforehand.

A good deal of exploratory work was carried out to determine useful ranges for the features. The more the features are allowed to vary, the wider the variety of faces produced. With large ranges, however, faces formed from extreme data can become very distorted. On the other extreme, too little variability in the ranges suppresses valuable information and hinders the clustering process.

It appears that the best ranges depend on the structure and amount of variability in the data. Every data set has its own characteristics, and it is best to tailor each to its own best set of ranges. A great portion of the SOF data is found "close" to the grand mean, but with a few

significant outliners. In order to provide discriminating ability among the largest mass of the data, the appearance of the outliers was further accentuated. The ranges were set at values which would allow close-in discrimination, but simultaneously attempted to minimize the departure of the outliers.

C. CLUSTERING THE FACES

The next step in this research is to cluster the faces. This task was performed by six students in the Operations Research curriculum. The faces are shown in figure 11; the 33 course segments from the Electrical Engineering department in quarter 781. The judges (students performing the clustering) were given no information concerning the feature-variable combination. They were simply instructed to group the faces in the manner which best suited them. Fifteen minutes were allowed for the task. The purpose was to quickly, but carefully, cluster the faces. The judges were reminded that each face is different and to search for the most natural looking clusters. It was felt that too much time spent on this task could defeat the purpose of the faces as a first pass look at the data. In every case, the judges acted independently of one another. No clues were provided which might have indicated which features were more important.

Figure 11 shows the faces in the clusters which were formed by the MIKCA algorithm. This cluster structure was

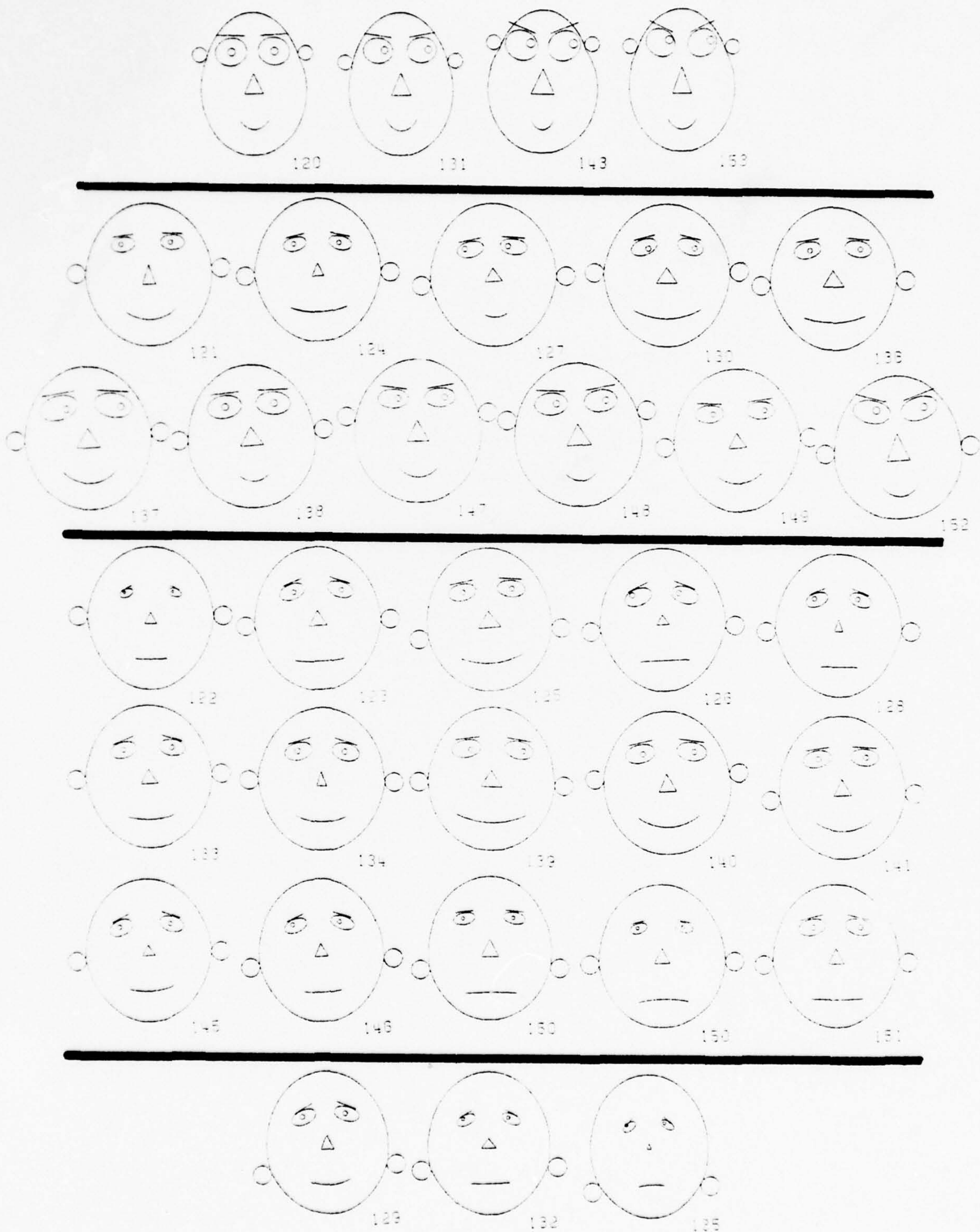


Figure 11

CLUSTERS DETERMINED BY MIKCA

used as a standard against which the judges' results were compared. Table 7 shows the results of this experiment. There was considerable agreement among the judges, as indicated by the comparison coefficients. There was also a good deal of similarity between the clusters formed by the judges and those formed by the MIKCA algorithm.

Several comments by the judges indicated the difficulties they encountered. The most prevalent comment was the difficulty in deciding which feature to consider the most important. One judge considered the mouth first in every case while another judge used the slant of the eyes as a more important variable. The judges also indicated that trying to evaluate simultaneously differences in many features was quite difficult. It is interesting to note that the judges' results were quite similar despite the fact that different criteria were employed as they formed the clusters.

The SOF identification numbers have been altered for this report. There were two course segments which erroneously reported the same SOF number (see face 150). As one looks at the faces with the discriminant space in mind, it is much easier to form a clustering which is similar to the MIKCA solution. One would be aware, for example, that the position of the pupils is critical in that it can diminish the effect of the smile and impact heavily on the horizontal dimension. This effect can be seen by referring to faces 139 and 140; they are included in a group whose smiles are

TABLE 7

COMPARISON COEFFICIENTS FOR ALL PAIRS OF JUDGES

JUDGES

	1	2	3	4	5	6
1	1.0	.69	.73	.82	.73	.78
2		1.0	.90	.80	.77	.68
3			1.0	.65	.81	.67
4				1.0	.68	.73
5					1.0	.69
6						1.0

COMPARISON COEFFICIENTS
BETWEEN EACH JUDGE AND MIKCA

JUDGE	COMPARISON COEFFICIENT
1	.59
2	.58
3	.68
4	.81
5	.76
6	.73

SIMULTANEOUS COMPARISONS OF
MULTIPLE JUDGES

NUMBER OF JUDGES	COMPARISON COEFFICIENT
3	.52
4	.44
5	.38

not as large as their own because the position of their pupils (positive to the right) has diminished the impact of the curvature of the mouth.

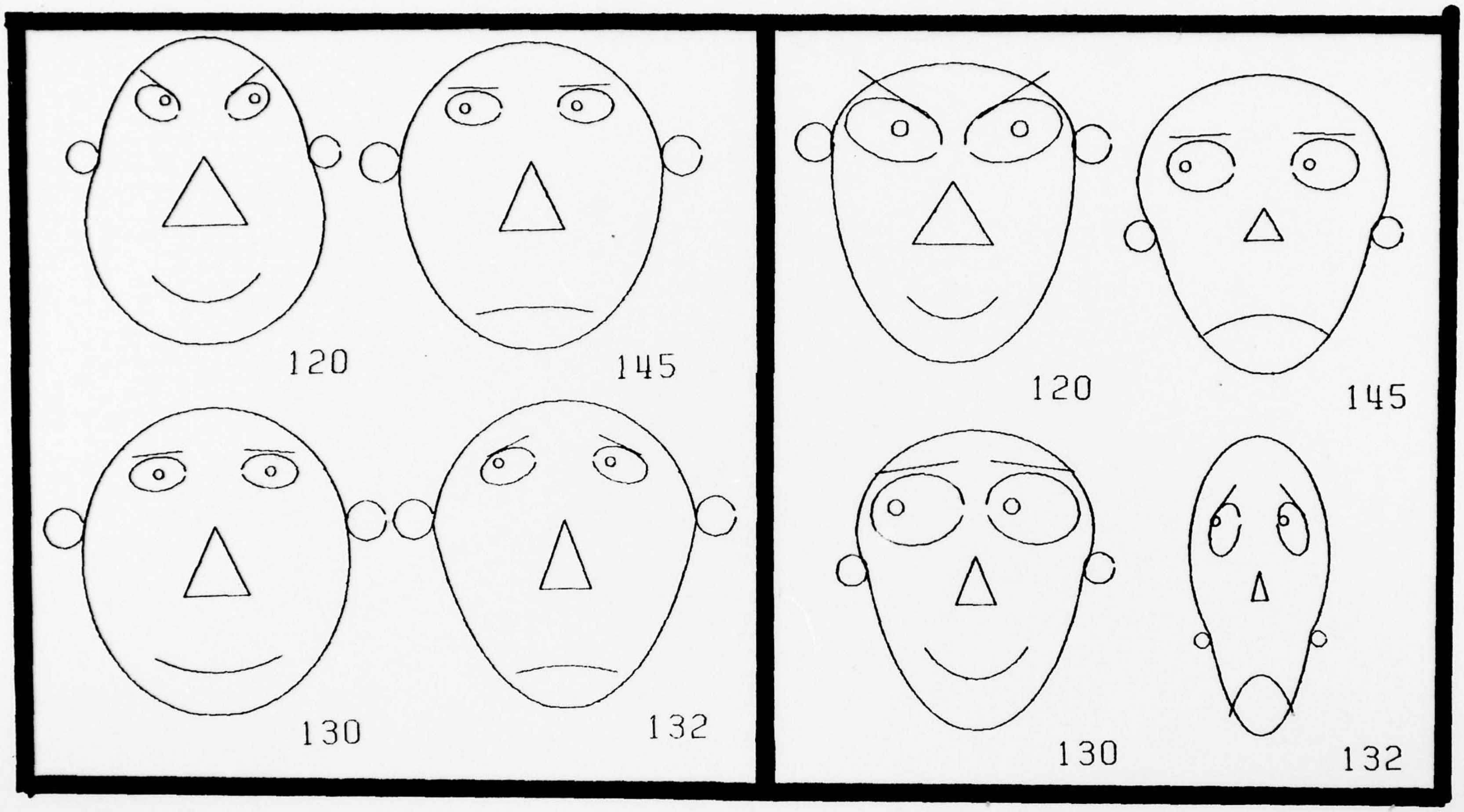
Another example of the interaction of variables is seen by referring to face 152. Most judges would quickly include this face with the group of four at the top of figure 11. A subtle difference, however, is the ear position. Reference to the discriminant function coefficients will indicate a negative which has offset the slant of the eyes in the second dimension.

Knowledge of the discriminant functions helps alleviate the confusion which sets in when attempting to cluster. It is especially true in this case where so little difference exists between the majority of the faces in the middle groups.

Difficulty in evaluating all 13 features simultaneously was a problem. As an alternative to this set of faces, two other sets were produced, one with only six variables features and the other with eight. Figure 12 contains samples from these sets, 12a the six variable set and 12b the eight variable. Of course, not all of the data is represented in this manner. Only those variables which loaded heavily in the discriminant analysis were used, and the features controlled by those variables are the ones considered to convey the most information. Table 6 gives the complete feature-variable combinations used in the construction of all sets of faces. The data used in constructing the set of 33 faces is found in Appendix B.

SIX VARIABLES

EIGHT VARIABLES



12a

12b

Figure 12

D. PROBLEMS ENCOUNTERED

The last section addressed difficulties faced by the judges because it was impossible for them to be aware of the information contained in the discriminant analysis. Of course, it would be pointed out that there is little reason to use this particular MIKCA solution as the standard, but it does serve as an objective standard, as it was desired to compare the machine results with the human results. This section addresses problems of a more mechanical nature.

Exploratory work with the faces uncovered quite a number of relationships between the features. The existance of geometric dependencies (not discriminant-type effects) between features caused difficulties in clearly displaying the variables. Two notable examples are mentioned here.

The length of the mouth is quite dependent upon its curvature. The projection on the horizontal axis (no relation to discriminant axis) has half-length

$$a_m = X9(h/|X8|)$$

where X8 is the mouth curvature. The variables which control these features are thus automatically forced into this dependent status regardless of their true relationship.

The other example concerns the ellipses forming the facial boundary and the angle theta. The upper ellipse is

drawn through the points P', U, and P; the lower through P', L, and P (see figure 10). Two faces with identical values for the ellipses might have quite different appearing facial boundaries due to the dependence on theta for the points P' and P. This is another example of forced dependence.

In order for the width and height of the face to meet a specified constant, the program "normalizes" both horizontal and vertical axes. This normalization eliminates the effects of X1 and X3, and it adjusts all of the features during the process. It is believed to be this normalization process which causes faces which are growing wider and wider to suddenly revert to one-half the widest width when the width exceeds a threshold value. A similar phenomenon is experienced in the height variable. This half-size adjustment may be seen in figure 12b. Face 132 has been changed by a disproportionate amount due to the normalization process. It is of interest to point out that the face-width feature was being held constant during the construction of that set of faces.

Yet another hidden dependency is that of the nose length on the eye height. The eyes are located at height

$$y_e = h[X10 + (1 - X10)X6]$$

where X6 controls the length of the nose.

These and other subtle dependencies mislead the investigator if he is not aware of their existence. These problems

reduce the ability of the faces to effectively display the full 20 dimensions. Unfortunately, these points are not explicit in the original document [11] and their discovery was an 11-th hour surprise. It was not possible to adjust for them or to uncover all such relationships at this writing. Appendix C gives a complete listing of the formulas used in the construction of the faces.

VII. COMPARISON COEFFICIENT

A. BACKGROUND AND ALTERNATIVES

Repeated use of the comparison coefficient has been made in this study. The present chapter is devoted to an explanation of this measure of association. The method should be flexible enough to handle multiple comparisons simultaneously, thus enabling one to measure the overall agreement of several judges.

It was decided the best way to display the agreement of two judges was through the use of a contingency table. Table 8 is an example of one to be used in the discussion.

Judge Y \ Judge X	A	B	C
A	5	0	1
B	1	3	3

Table 8

The contingency table indicates the agreement of the two judges. The purpose of this chapter is to find a measure which evaluates how close this agreement is. Note that judge X categorized the observations into three clusters with 6, 3, and 4 elements, respectively. Of the 13 observations, judge Y placed six in one group and seven in another.

The labeling of the clusters is arbitrary. The upper left entry in the table indicates that five of the objects in judge X's category A matched with five of judge Y's category A. The entire table is interpreted in this manner. Notice that if one chooses to call this entry of five as representing agreement, then the entry of 1 below it and the 1 in the top right corner must represent some of the observations on which the judges disagreed.

The contingency table idea is easily generalized to higher dimensions (more than two judges). In three dimensions, a box (or cube) would represent the table, with elements internal to the box measuring agreement between three judges.

One method for measuring the degree of agreement is to find the largest combination of entires such that only one per row and one per column are chosen. This task becomes very difficult as the number of clusters increases, but it can be solved through the use of linear programming techniques. It is a constrained optimization with a linear objective function and is an application of the "assignment problem." Unfortunately, when generalizing to higher dimensions, the L.P. loses its unimodularity attribute and the number of constraints and variables in the problem becomes prohibitively large.

The Chi-square contingency statistic was considered inappropriate because, when using the smaller sample sizes, more than 20 percent of the cells have expected frequencies

of less than five (see ref. 14). Even when using the 190 element sample, there were frequent occasions when this same difficulty persisted. The Chi-square statistic was not used since it could not have been applied consistently throughout the analysis.

Professor James Hartman provided an idea that led to the method finally put into use.

B. THE TECHNIQUE

The idea was to sum the squares of the entries in the contingency table and then "normalize" this quantity. Summing squares offers an excellent method for measuring the degree of association, however the following example illustrates the need for some sort of adjustment factor.

10	0
0	10

9a

19	0
0	1

9b

TABLE 9

Both tables represent perfect agreement on 20 observations, however table 9a has a sum of squares equal to 200 and 9b has a value of 362. It is desired to indicate both of these examples as perfect agreement with one being no better than the other. Hence, it became necessary to determine the "best possible" sum of squares in every given

situation. A computer program was written for this purpose and is included in Appendix F. The statistic which is used as a comparison coefficient is a number between zero and one, formed as the ratio of the actual sum of squares to the "best possible" sum of squares. The best possible sum of squares is a computed sum using a minimax approach and is based on the number of judges, number of clusters by each judge, and the number of observations within each cluster. The minimax procedure does not need to know which observations make up a cluster, only how many observations. An example showing the computation of the comparison coefficient is given in Appendix E.

This method for measuring the degree of agreement provides the analyst a standard scale upon which to compare coefficients based on solutions involving varying numbers of clusters and cluster memberships, as well as varying numbers of judges.

In order to provide some sensitivity for the significance of this measure, several cluster solutions were formed wholly at random and compared to results produced by MIKCA and the judges. In every case, the values of the comparison coefficients were less than 0.1.

VIII. SUMMARY AND CONCLUSIONS

This research has been largely exploratory. A path has been paved for others to follow in examining the SOF data. The theory of cluster analysis and its relationship to discriminant theory have been carefully examined with emphasis on two widely divergent techniques. In the analysis of the data, attempts have been made to identify the underlying structure of which the clusters are a consequence. This chapter is devoted to separate discussions of the cluster methodology explored and the interpretation of the SOF data.

Although the development of methodology phase of the research was carried to completion in a general sense, a number of problems were encountered along the way. Many of these problems are deserving of deeper treatment and are discussed below.

The data transformation was the best of the three considered. It produced the smallest test statistic for homogeneity of covariances, but the value itself was not in the acceptable range, based on normal theory. It should be possible to improve the choice.

The modifications of MIKCA to allow for weighting of the input vectors has been effected and well tested. It is an important added capability for this program.

The use of discriminant analysis to discover the important variables affecting the clustering is, no doubt,

not new. It needs some refinement, however, because it is not clear how one should rate the importance of variables supporting the first dimension to those supporting the second (or any other dimension). Such a set of priorities could be most useful.

The idea of using the important variables (and their signs) of the discriminant functions in the problem of assigning sets of variables to sets of features is believed to be new. It may have great potential in providing a way for the Chernoff face technique to replace the more expensive technique based on computer iteration.

The present attempt to work with the faces was disappointing. This is due largely to the face that certain restrictions, truncations, and discontinuities in the movement of the features were not well documented in our sources. Their discovery came as a surprise late in the research and it was not feasible to go back and readjust. Such readjustment is clearly called for and would require a substantial effort in the future studies.

The coefficient of comparison was a new idea and there was insufficient time to explore its properties. What is needed is more investigation in order to interpret its various values (or another measure whose values are interpretable). The comparison measure is also useful in the problem of assigning variables to features when working with faces. The goal is to choose assignments having the property

that the judges are in good agreement when forming the clusters.

The study of the SOF data which was made while developing the cluster methods produced results about student evaluations of courses (and instructors). The results are discussed below.

A principal components analysis of the data swarm of mean vectors showed it to be essentially one dimensional and having the direction of the main diagonal of 13-space. The interpretation os this is that all 13 items are equally important in the students' perception of rating the course and its instructor. On the other hand, this same effect would be produced by careless, perfunctory completion of the forms by many students.

The partitioning of the data into three or four clusters by MIKCA is more or less successful. The clusters are not sharply separated (there are no great voids between them). Study should be made to see how much the density of the data diminishes near the boundaries of the partitions.

Although the main data swarm is essentially one dimensional, it appears useful to use two dimensions to describe the individual partitions after clustering. In doing this, variable 12 (overall rating of the instructor) emerged as most important in the first dimension and variables 3, 8, and 11 giving support in the second. Only one discriminant study is reported here, although several were performed.

Variable 12 appears to have permanence while the other variables often shift in importance.

The cluster profiles which track the cluster centroids over the 13 variables provide a set of (almost) horizontal lines. This supports further the one dimensional interpretation of the data swarm. The result is not sensitive to whether or not the data are standardized.

The results of applying the modified MIKCA did not vary greatly from the results of applying the original MIKCA. Hence the number and composition of the clusters is not disturbed much by the variability in class size.

The relative position of the clusters is strongly and inversely related to class size. The courses that receive uniformly high ratings are associated with the small class sizes and the courses receiving uniformly poor ratings are associated with the large class sizes.

All judges reported use of a hierarchical approach to separate the faces into clusters. Most judges first separated the faces into two groups according to the curvature of the mouth (smile or frown). There was little agreement about which features were important in further subdividing the two main groups, hence some disparity resulted in their final cluster solutions.

The MIKCA procedure is a sophisticated approach to cluster analysis; its results are based on sound statistical theory. The modified version of that computer program is considered

particularly well suited to the SOF data or any other data set possessing the same predetermined class structure. The impact of class size on cluster membership has been emphasized. This important issue may indicate the smaller classes receive artificially inflated SOF scores. Consideration to this fact surely must be given by those who use these scores as a means for evaluating teacher performance.

APPENDIX A

Test for the Equality of Dispersion Matrices of k Groups

Given a sample of k groups and m variables with group dispersion matrices, S_i, (i = 1, ..., k) pooled within-groups dispersion matrix S_w, and total sample observations $N = \sum_{g=1}^{k} N_g$, Box shows that the hypothesis

$$H_1: \Sigma_1 = \Sigma_2 = \dots = \Sigma_k$$

may be tested by an F statistic developed from

$$A = \ln[|S_w|]\cdot[N - k] - \sum_{i=1}^{k} (N_i - 1)\cdot \ln(|S_i|)$$

$$B = \frac{\left[\sum_{i=1}^{k} \frac{1}{1 + N_i} - \frac{1}{N - k}\right] \cdot (2m^2 + 3m - 1)}{6(k - 1)(m + 1)}$$

$$C = \frac{\left[\sum_{i=1}^{k} \frac{1}{(1 + N_i)^2} - \frac{1}{(N - k)^2}\right] \cdot (m - 1) \cdot (m + 2)}{6(k - 1)}$$

$$D = \frac{(k - 1) \cdot m \cdot (m + 1)}{2}$$

$$E = \frac{D + 2}{\text{abs}\ B^2 - C}$$

If $B^2 > C$, the test statistic is

$$\left(\frac{E}{D}\right)\left[\frac{A(1 - B + 2/E)}{E - A(1 - B + 2/E)}\right] \sim F_E^D$$

If $C > B^2$, the test statistic is

$$\left(\frac{A}{D}\right)(1 - B - D/E) \sim F_E^D$$

AD-A068 544 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 12/1

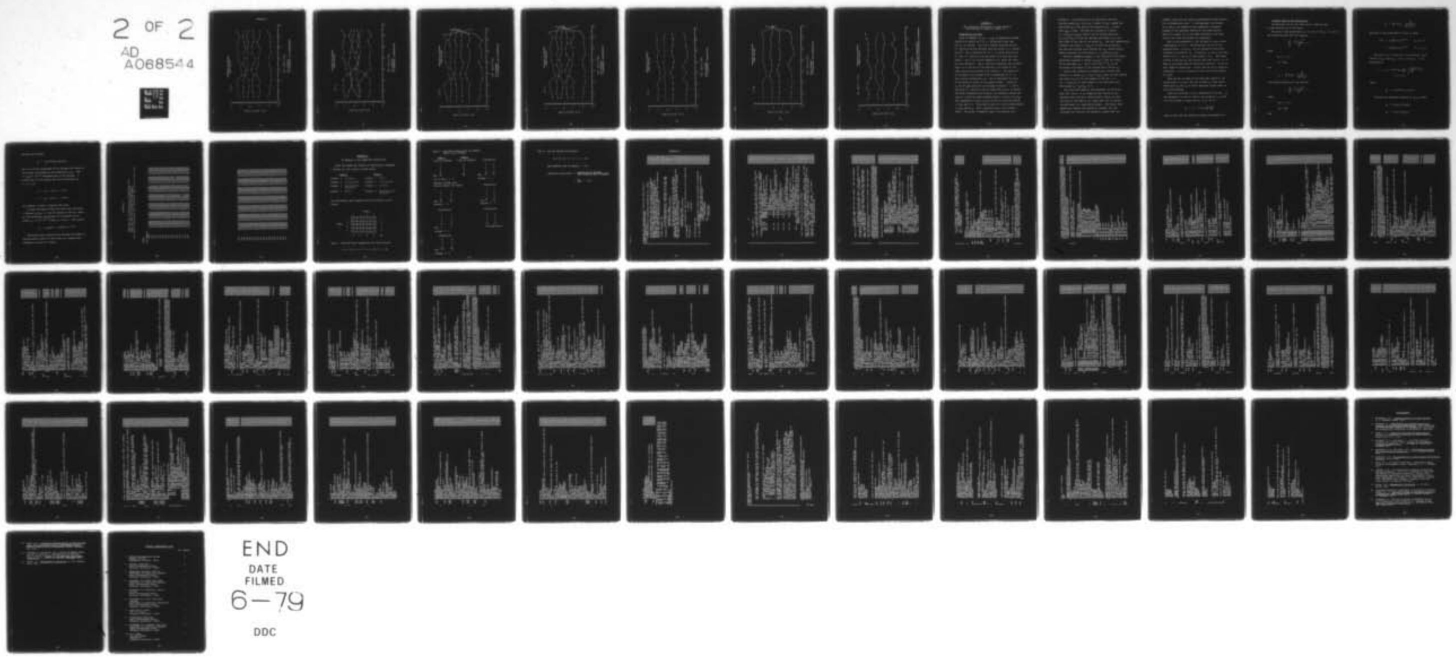
DEVELOPMENT OF CLUSTER ANALYSIS METHODS SUITABLE FOR STUDENT OP--ETC(U)

MAR 79 J W AIKEN

UNCLASSIFIED NL

2 OF 2

AD A068544



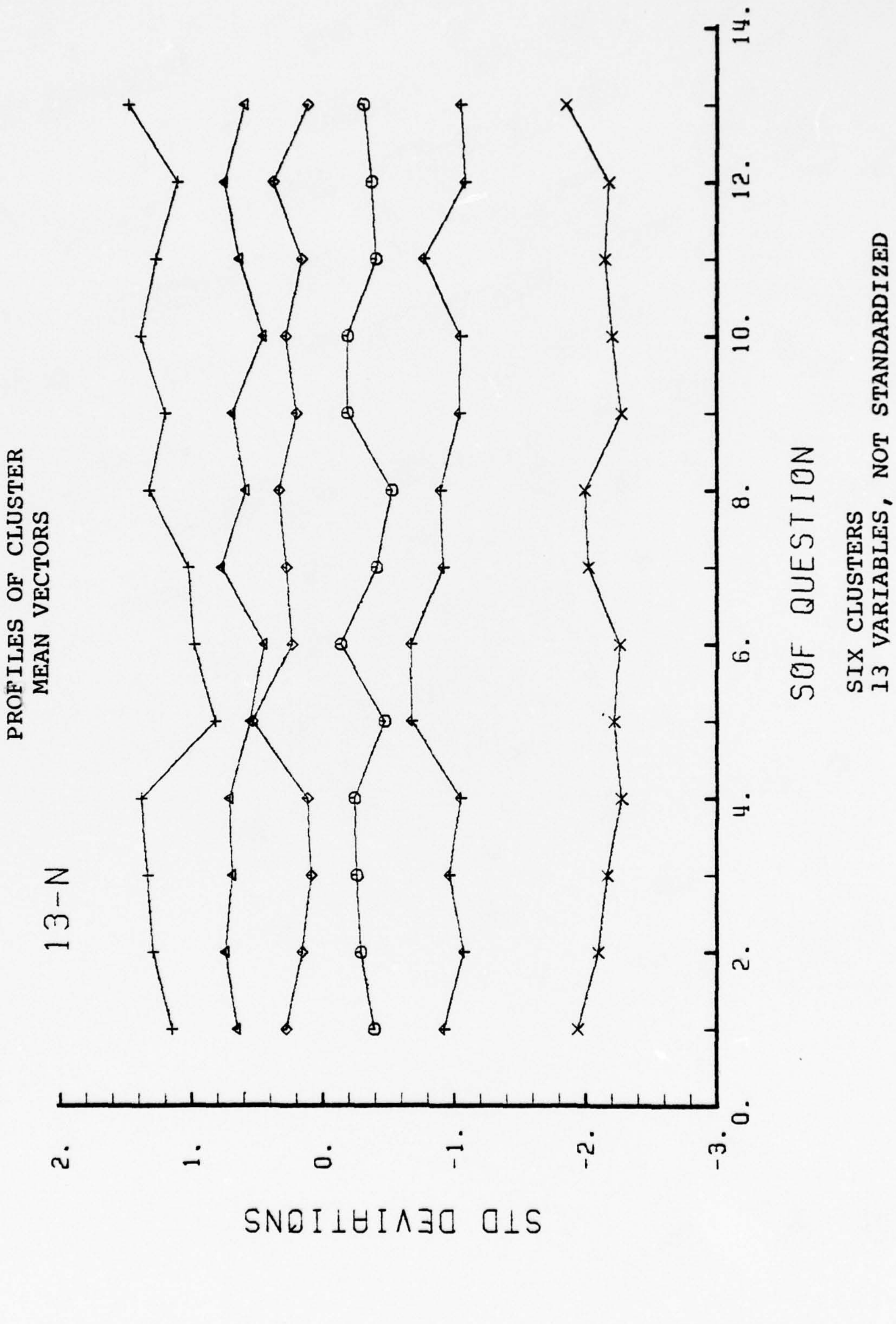
END

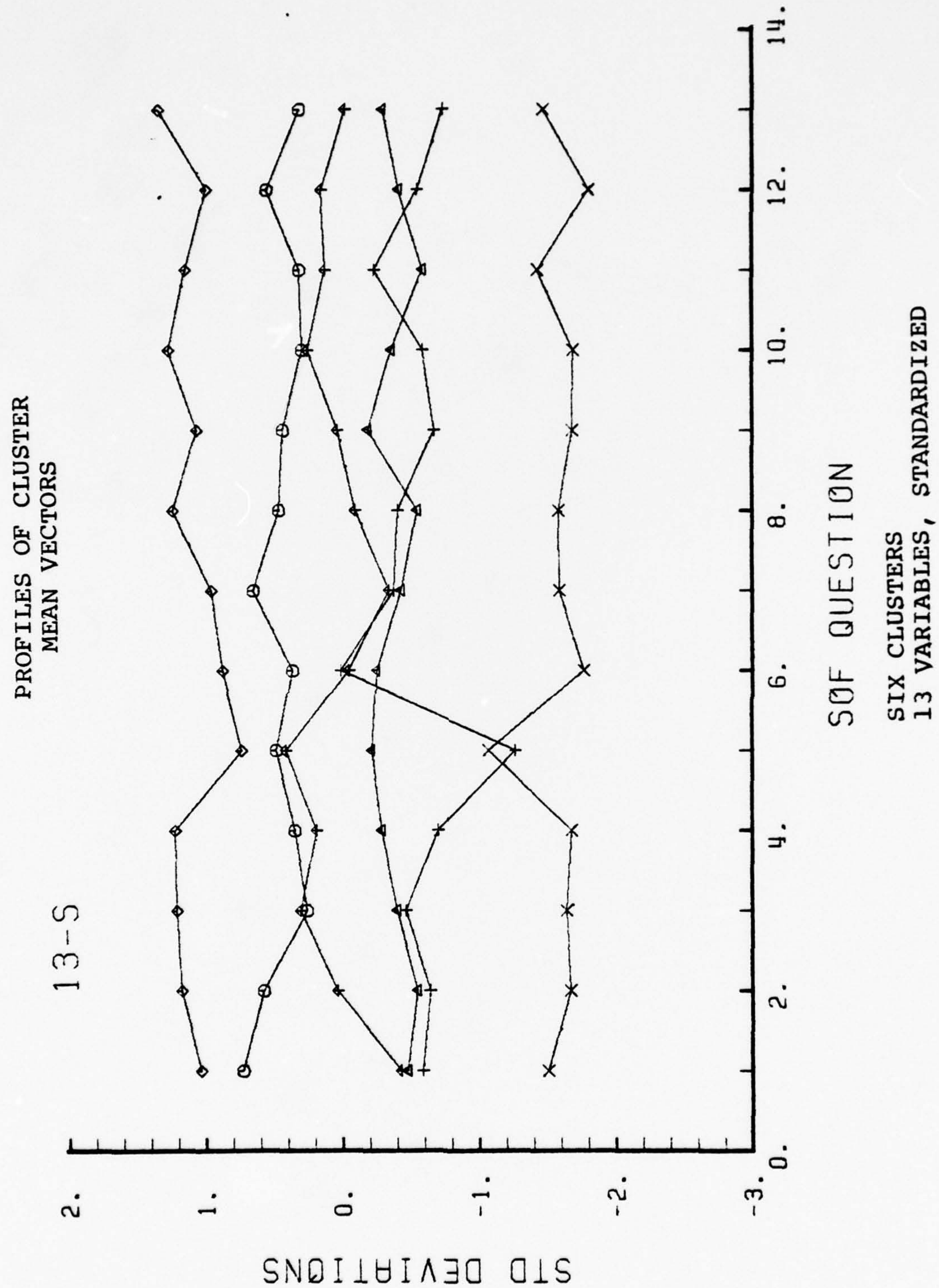
DATE FILMED

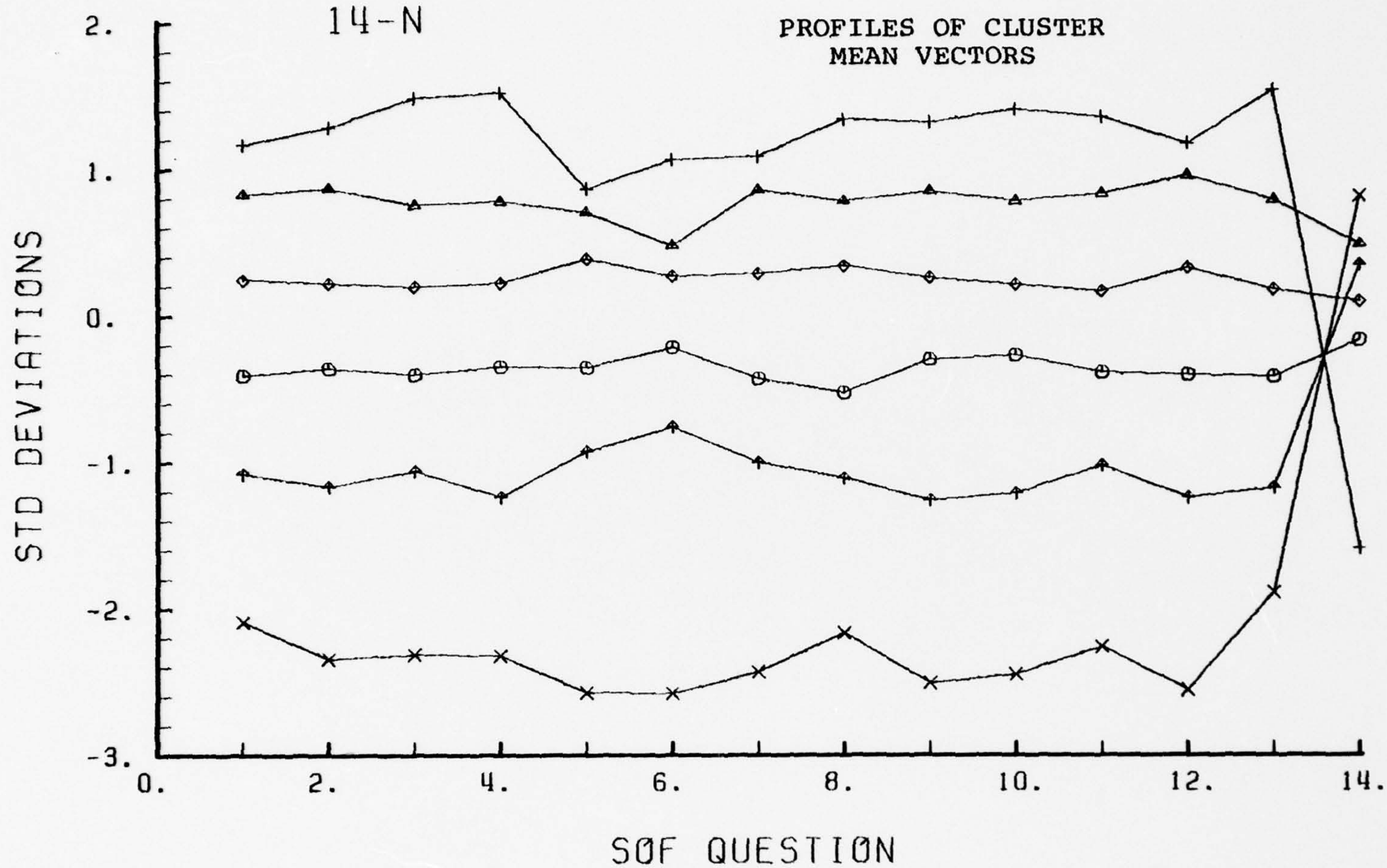
6-79

DDC

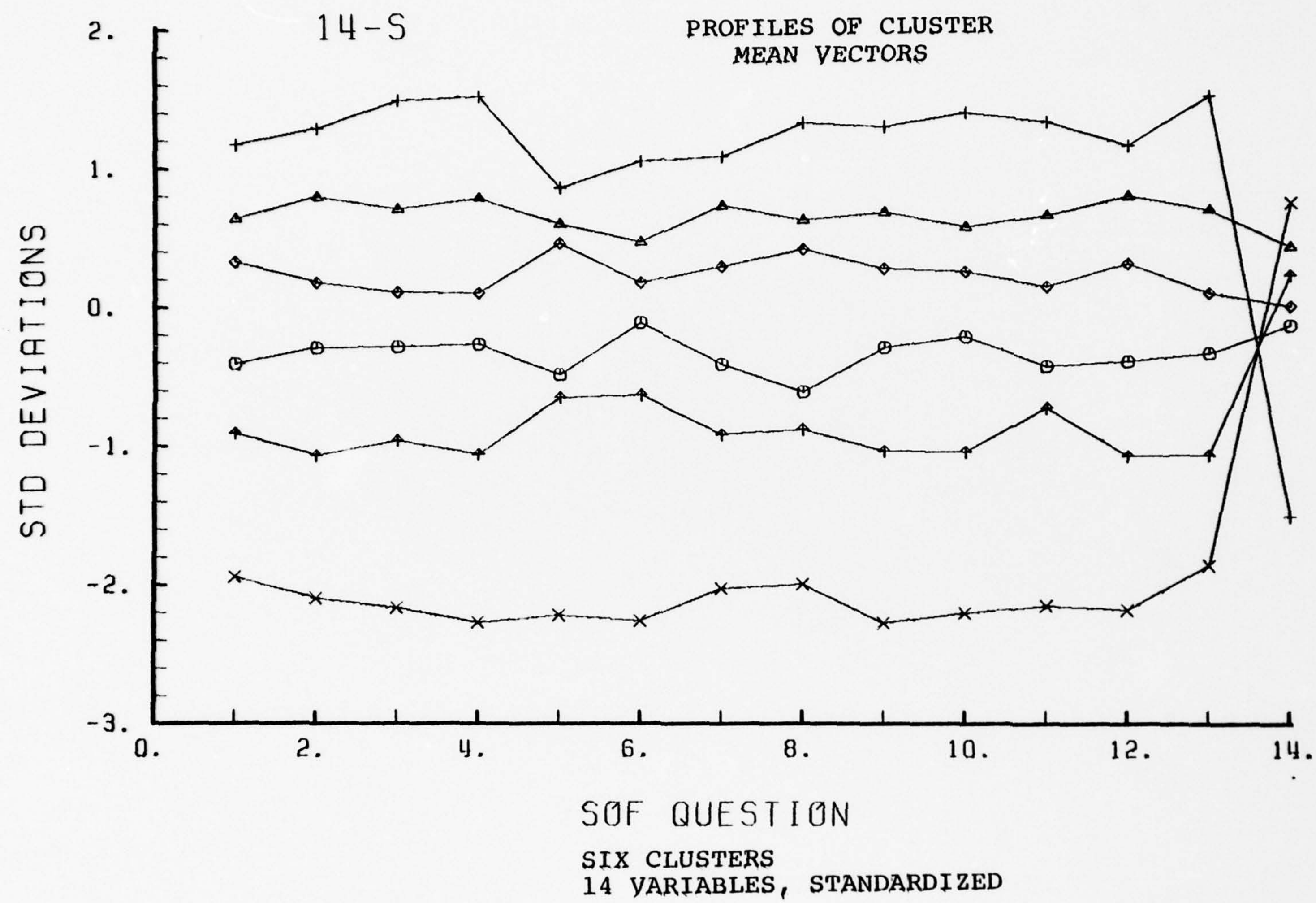
APPENDIX B


PROFILES OF CLUSTER
MEAN VECTORS
13-N
STD DEVIATIONS
2.
1.
0.
-1.
-2.
-3.
0.
2.
4.
6.
8.
10.
12.
14.
SØF QUESTION
SIX CLUSTERS
13 VARIABLES, NOT STANDARDIZED

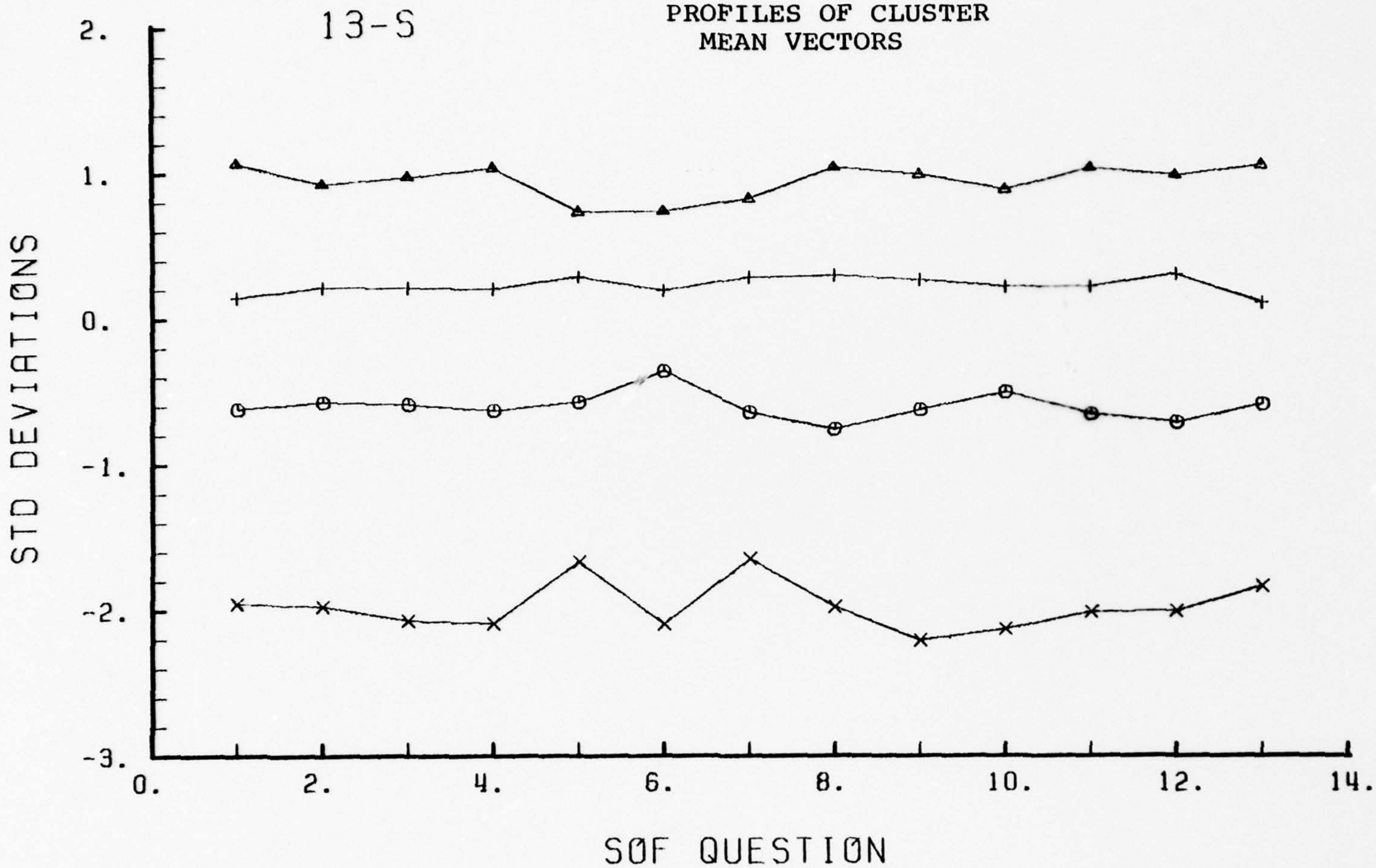

PROFILES OF CLUSTER
MEAN VECTORS
13-S
STD DEVIATIONS
2.
1.
0.
-1.
-2.
-3.
0.
2.
4.
6.
8.
10.
12.
14.
SØF QUESTION
SIX CLUSTERS
13 VARIABLES, STANDARDIZED



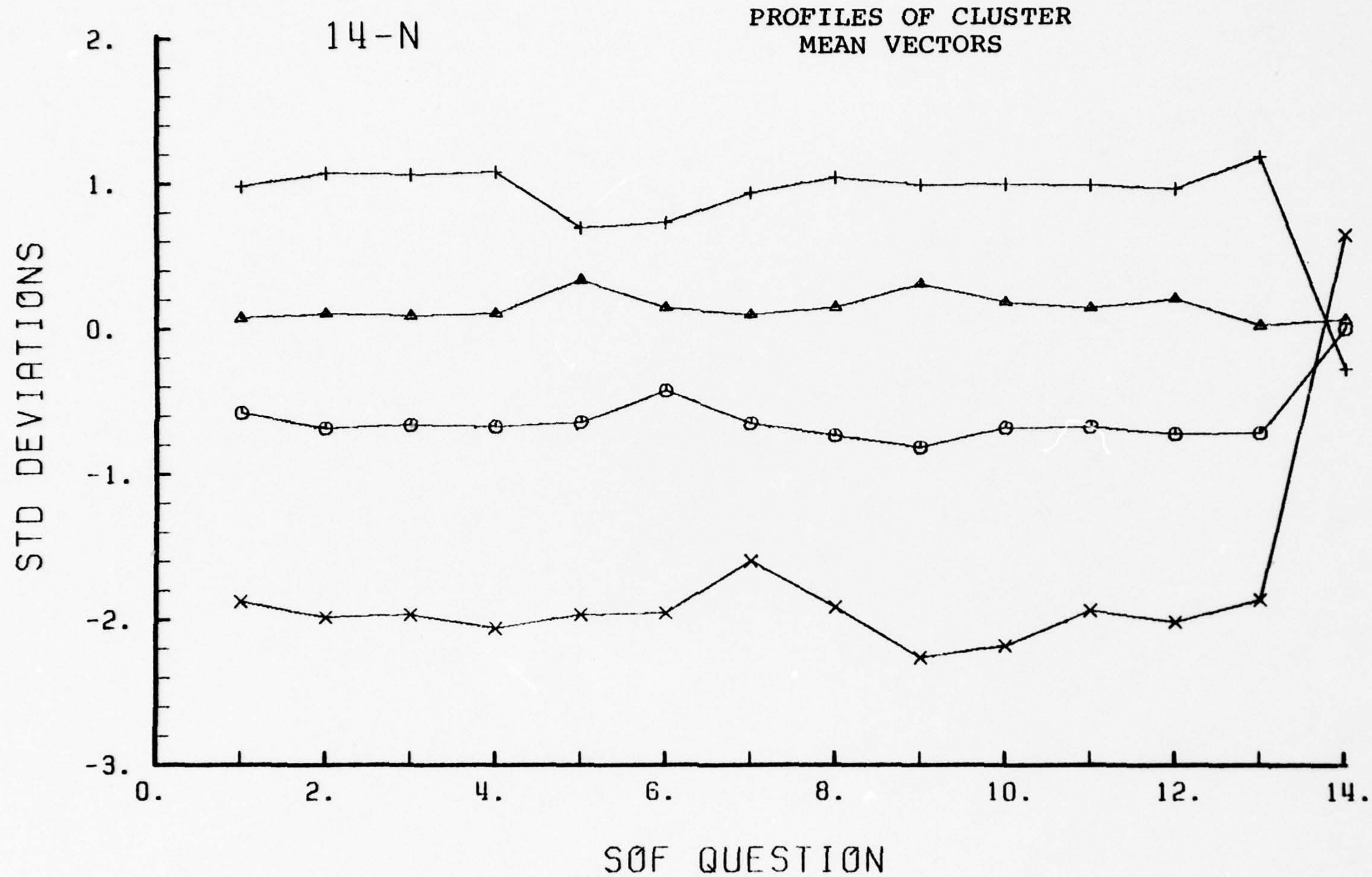
SIX CLUSTERS
14 VARIABLES, NOT STANDARDIZED


14-S
PROFILES OF CLUSTER
MEAN VECTORS
STD DEVIATIONS
2.
1.
0.
-1.
-2.
-3.
0.
2.
4.
6.
8.
10.
12.
14.
SØF QUESTION
SIX CLUSTERS
14 VARIABLES, STANDARDIZED

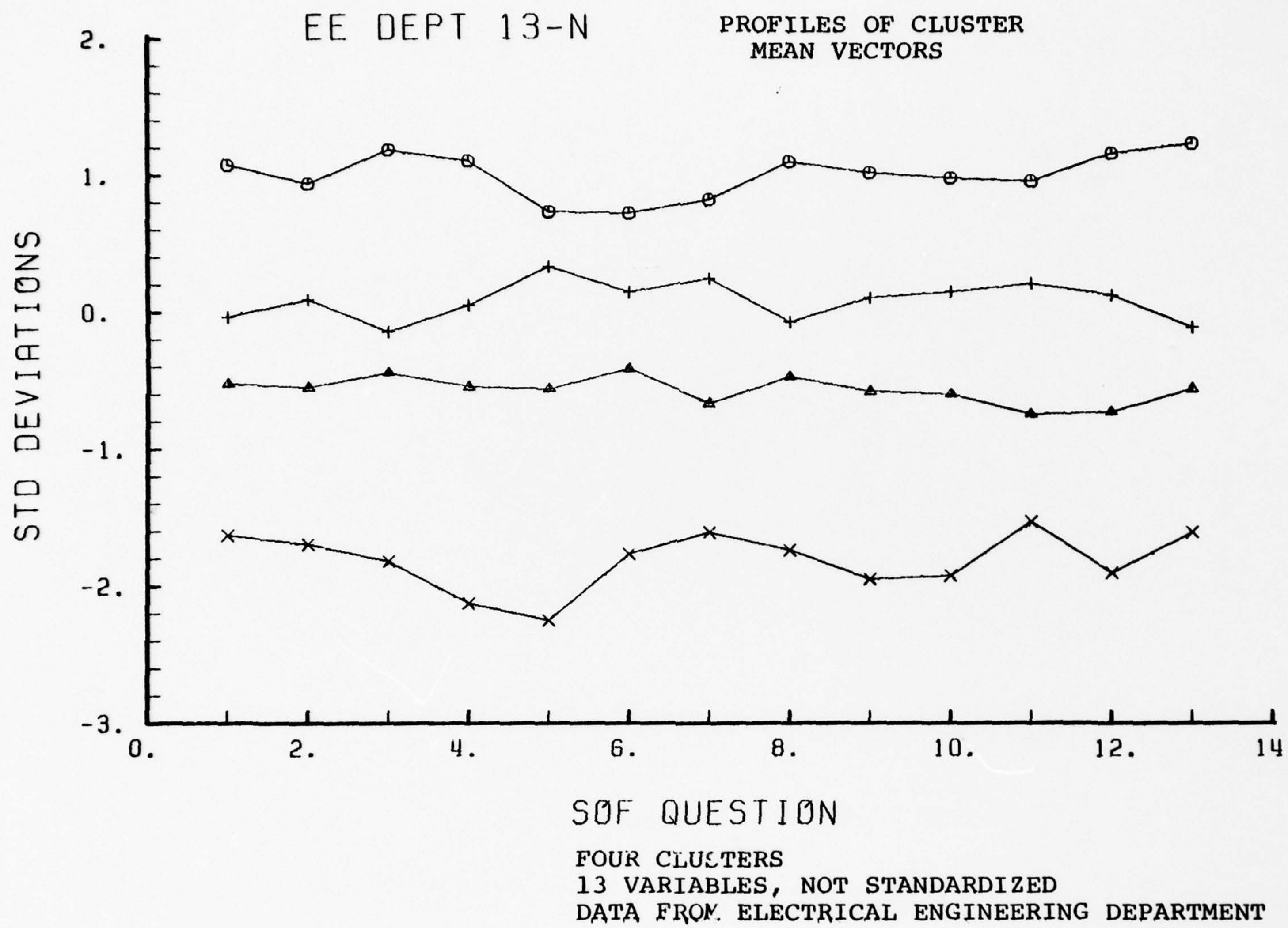
PROFILES OF CLUSTER
MEAN VECTORS



FOUR CLUSTERS
13 VARIABLES, STANDARDIZED



FOUR CLUSTERS
14 VARIABLES, NOT STANDARDIZED



EE DEPT 13-N
PROFILES OF CLUSTER
MEAN VECTORS
STD DEVIATIONS
2.
1.
0.
-1.
-2.
-3.
0.
2.
4.
6.
8.
10.
12.
14.
SØF QUESTIØN
FOUR CLUSTERS
13 VARIABLES, NOT STANDARDIZED
DATA FROM ELECTRICAL ENGINEERING DEPARTMENT

APPENDIX C

The following information is taken directly from Chernoff's technical report [9]

Construction of Faces

Given 18 numbers $(x_1, x_2, \ldots, x_{18})$ in appropriate ranges (which will usually be 0 to 1), we define a face (see Fig. 3) as follows. Let H be a nominal distance and let $h^* = \frac{1}{2}(1+x_1)H$ be the distance from the origin to a "corner" point P. As x_1 varies from 0 to 1, h* varies from H/2 to H. Let $\theta^* = (2x_2-1)\pi/4$ be the angle of OP with the horizontal. Let P' be a point symmetric to P about the vertical axis through O. Let $h = \frac{1}{2}(1+x_3)H$ represent the distance from O to U the top of the head and L the bottom of the head, both on the vertical line through O. The upper part of the head is an ellipse which is determined by P', U, and P and an eccentricity x_4. Let x_4 represent the ratio of the width to height of the upper ellipse. Similarly, x_5 is the same ratio for the ellipse through P', L, and P. The nose is a vertical line of length $2hx_6$ with O as center. The mouth intersects the vertical line extended through the nose at a point P_m whose distance below O is $h[x_7+(1-x_7)x_6]$. This represents a point x_7 part of the way from the bottom of the nose to U. The mouth is part of a circle whose center is h/x_8 above P_m. Thus a positive value of x_8 yields a smile. The mouth is symmetric about the vertical axis

through 0. Its projection on the horizontal axis has the half-length $a_m = x_9(h/|x_8|)$ unless $(h/|x_8|)$ exeeds the half-width w_m of the face at the height of P_m. In that case $x_9 w_m$ is used. The eyes are located at a height $y_e = h[x_{10}+(1-x_{10})x_6]$ above 0 and at centers which are $x_e = w_e(1+2x_{11})/4$ from the vertical axis where w_e is the half-width of the face at the height y_e. They are symmetrically slanted at an angle $\theta = (2x_{12}-1)\pi/5$ with the horizontal. The eyes are ellipses with eccentricity x_{13} (height/length before slanting) and half-length $L_e = x_{14}\min(x_e, w_e - x_e)$.

The only asymmetry appears in the location of the pupils which move together an amount $r_e(2x_{15}-1)$ from the center of the eye where $r_e = (\cos^2\theta + \sin^2\theta/x_{13}^2)^{-1/2}L_e$ is the horizontal half-length of the slanted eye at height y_e.

Finally the eyebrows are symmetrically located with centers at a height $y_b = 2(x_{16}+.3)L_e x_{13}$ above the eye centers and slant $2(x_{17}-1)\pi/5$ with respect to the eye, i.e., $\theta^{**} = \theta+(2x_{17}-1)\pi/5$ with respect to the horizontal and half-length $L_b = r_e(2x_{18}+1)/2$.

One final step taken by the programmer and which has been left intact, is to normalize both horizontal and vertical axes, each by a multiplicative factor, so that the width of the head at its widest part and its height are both equal to a specified constant. This step, which essentially removes two degrees of freedom, was left unaltered for intuitive and aesthetic reasons that are

somewhat vague and may require reconsideration when dealing with 18-dimensional data. In the meantime, the effects of x_1 and x_3 are almost but not completely eliminated because of the secondary effects of the normalization, which will adjust all of the other features at the same time as the width and height are normalized.

Most of the parameters x_i are adjusted to range within a subinterval of (0,1). The exceptions are two of the eccentricities, x_4 and x_5, and the parameter controlling curvature of the mouth, x_8. Ordinarily, x_4 and x_5 are kept within 1/2 to 2, and x_8 is kept within (-5,5). The eccentricity of the eye x_{13} has usually been kept within (.4,.8). Some of the ranges must be controlled carefully. We do not want negative length eyes. Others need not be so carefully controlled. It is no calamity to have eyes extend beyond the face.

When the two ellipses of the head meet smoothly, the corner point P is lost, and the variable x_2 loses effect. Restricting x_4 and x_5 to widely separated ranges seems to avoid this problem.

Data are converted to the x parameters as follows. If the variable Z is used to control the parameter x_i, which is to be allowed to range from a_i to b_i, we let

$$x_i = a_i + (b_i - a_i)\left|\frac{Z - m}{M - m}\right|$$

where m and M are the observed minimum and maximum of Z.

Formulae Used on the Construction

We describe a few of the less trivial formulae used in the construction of the faces.

The point P has coordinates $x_o = h^* \cos \theta^*$ and $y_o = h^* \sin \theta^*$. The ellipse through PUP' has equation

$$\frac{x^2}{a_u^2} + \frac{(y - c_u)^2}{b_u^2} = 1$$

where

$$b_u = h - c_u ,$$

$$a_u = x_4 b_u$$

and

$$c_u = \frac{1}{2}[(h+y_o) - \frac{x_o^2}{x_4^2(h-y_o)}] .$$

The ellipse through PLP' has equation

$$\frac{x^2}{a_L^2} + \frac{(y - c_L)^2}{b_L^2} = 1$$

where

$$b_L = h + c_L ,$$

$$a_L = x_5 b_L$$

and

$$c_L = \frac{1}{2}[(-h+y_o) - \frac{x_o^2}{x_5^2(-h-y_o)}]$$

The head is then described by $(\pm x(y),y)$ where

$$x(y) = x_4[b_u^2-(y-c_u)^2]^{1/2} \qquad y_o \leq y \leq h$$

$$= x_5[b_L^2-(y-c_L)^2]^{1/2} \qquad -h \leq y \leq y_o$$

The mouth is a circular arc with curvature $|x_8/h|$ through $(0,y_m)$ where $y_m = -h(x_7 + (1-x_7)x_6)$. It is described by

$$y = y_m + (\text{sgn } x_8)[\frac{h}{|x_8|} - \sqrt{(\frac{h}{x_8})^2 - x^2}],$$

$$0 \leq x \leq a_m$$

where

$$a_m = x_9 \min[x(y_m), h/|x_8|] .$$

The eyes are nominally centered at (x_e,y_e) where

$$y_e = h[x_{10}+(1-x_{10})x_6]$$

$$x_e = x(y_e)[1+2x_{11}]/4$$

and have half-length

$$L_e = x_{14} \min[x_e, x(y_e) - x_e] .$$

Let (u,v) be the coordinates of an ellipse with center at the origin, half-length L_e and eccentricity x_{13}. Then $v = x_{13}(L^2 - u^2)^{1/2}$ describes part of the ellipse. A similar part of the slanted eye can be described for $0 \leq u \leq L$ by

$$x = x_e + u \cos\theta - v \sin\theta$$

$$y = y_e + u \sin\theta + v \cos\theta$$

and symmetry is used to complete both eyes.

To place the pupils within the eyes, both are moved a distance $r_e(2x_{15} - 1)$ from the center of the eye, where r_e, the horizontal half-length of the slanted eye at height y_e, is $(u^2 + v^2)^{1/2}$ when $v/u = \tan\theta$. This yields

$$r_e = L_e(\cos^2\theta + x_{13}^{-2} \sin^2\theta)^{-1/2}$$

The program then normalizes all heights and widths by multiplicative factor k/h and k/max x(y) respectively. Currently k is set at 2 inches.

APPENDIX D

These are the 33 observations (transformed data) from the Electrical Engineering Department.
There are two rows of data per SOF number.
The first 13 entries are SOF item scores, the 14-th is class size.

CODED SOF NUMBER							
120	2.2896	1.0676	2.3623	2.3412	2.9299	2.7006	2.3982
	2.3982	1.9121	1.5421	2.1931	1.9037	1.5053	2.9444
121	1.5095	0.5676	1.1238	1.2215	1.7547	1.2643	1.1289
	1.3418	1.6801	1.3418	1.5095	1.1417	1.0148	2.1972
122	0.9161	0.1635	0.3354	1.1613	0.9076	0.1720	0.9994
	0.7441	0.3354	0.0817	0.7526	-0.4989	0.1635	2.4849
123	1.0978	1.2854	0.6427	0.9102	1.4638	1.6421	1.7405
	1.0179	0.9288	0.3752	0.8210	0.4644	0.1783	2.3979
124	0.5702	0.7388	0.6579	0.9083	2.0151	1.4770	0.9561
	0.8051	0.6477	0.8000	1.0486	0.8650	0.5235	3.0445
125	1.4376	1.1388	0.7079	0.4904	1.8798	1.7675	2.3148
	1.4376	1.2202	1.2202	1.0798	1.0666	0.8774	2.8332
126	0.6866	-0.1876	0.9032	0.7319	0.9288	1.9189	1.1197
	0.3567	0.0799	0.3567	0.5443	0.5006	0.1783	2.3979
127	0.2342	-0.0701	0.5678	0.8259	2.2508	2.0973	1.6478
	0.5678	0.6947	1.0796	2.1753	0.3430	0.1474	2.6391
128	0.1820	0.3736	0.6636	0.8037	1.3369	1.5286	0.5138
	0.1353	0.4155	0.1820	0.5653	-0.1498	-0.1031	3.0445
129	0.4099	0.6467	0.2885	-0.0637	0.5367	1.8964	1.1226
	0.8774	0.7983	0.2945	1.3053	0.0060	0.0	2.8332
130	0.7629	1.9094	0.7742	1.2215	2.8152	1.7890	2.1273
	0.8832	1.0993	0.2942	1.1125	1.5121	0.0976	2.3026
131	2.5859	2.1273	2.1273	2.0229	2.6318	2.3566	2.4817
	2.2476	2.2476	2.1387	2.2476	2.4817	1.3653	2.4849
132	0.5830	0.0517	-0.2368	0.0	0.8774	0.9531	1.3805
	0.1791	0.0485	-0.0697	0.7164	-0.1274	-0.3522	2.8332
133	1.1577	0.9334	0.9334	1.0673	0.6412	1.6042	1.6294
	0.9772	0.9133	0.8060	1.0212	0.3195	0.4635	3.8712
134	0.8991	0.7904	1.2816	1.0035	1.3418	1.8551	0.6709
	0.9044	0.8937	0.6818	1.2195	0.5323	0.6302	2.9444

135	-0.5794	-0.3362	-0.2548	-0.7709	-0.9144	0.7145	0.0613
	-0.1455	-0.6238	-0.7944	0.0923	-0.8508	1.1692	3.5264
136	1.5711	1.4508	0.8832	0.8832	2.5859	1.5711	2.1273
	0.6652	1.2328	0.8945	0.9922	1.2215	0.3269	2.1972
137	2.4549	2.0200	1.1647	1.1501	3.0445	2.5286	2.4549
	1.4303	1.7252	0.8407	1.5850	1.7252	0.9936	2.1972
138	1.4394	1.1011	1.6687	1.1125	2.3566	2.5286	2.3566
	2.1273	1.6687	1.4394	2.3566	1.6321	0.8710	2.1972
139	1.9217	1.7936	0.8629	0.8597	2.4315	1.7004	2.0667
	1.7558	1.2693	0.9302	1.0283	1.5049	0.6798	3.7612
140	1.7359	1.4352	0.9536	1.3598	1.8114	1.8604	1.6269
	1.7624	1.4400	1.0440	1.3104	1.0503	0.5350	3.1555
141	1.1011	1.2215	0.7855	0.6652	2.2706	1.8980	1.6801
	1.3418	1.2328	1.2328	1.3418	0.8945	1.2328	2.1972
143	2.7497	2.4549	2.7497	2.7497	3.0445	3.0445	3.0445
	2.7497	2.7497	2.7497	3.0445	2.7497	2.4549	2.0794
145	1.4150	0.6334	1.1553	0.8259	1.3062	1.2430	1.0796
	0.8259	0.2185	0.1430	1.0798	0.1550	0.0817	2.5649
146	0.9387	0.2452	0.1903	0.5645	1.6385	1.3869	1.1966
	0.9387	0.6258	0.2516	0.9450	-0.0064	0.1839	2.8332
147	1.1162	1.7547	0.9936	1.7547	3.0445	2.2706	2.2706
	1.7547	2.0127	2.0127	2.0127	1.7547	0.9936	2.0794
148	1.5850	1.8025	1.8025	1.8798	2.8971	2.3075	2.5322
	1.8726	1.9499	1.8726	2.4549	1.9499	1.2975	2.6391
149	1.6687	1.6687	0.9808	0.9808	3.0445	1.6687	1.6687
	0.6539	0.9808	1.6687	1.6687	1.3418	0.6539	1.0986
150	0.6968	0.6968	0.2942	0.8051	2.8381	0.7172	1.4140
	1.6101	1.0012	0.5006	0.2942	0.3923	0.2675	2.3979
150	1.7184	0.7070	-0.1962	0.2064	2.8381	1.0012	1.7082
	1.4140	1.3159	1.0012	0.5006	0.6089	0.5987	2.3026
151	1.0035	1.2328	0.7742	0.7484	1.6321	1.8004	1.3863
	0.5676	1.1417	0.8945	0.6652	0.6766	0.5562	2.1972
152	1.6687	1.6687	0.9808	0.9808	2.3566	3.0445	2.3566
	2.3566	3.0445	3.0445	2.3566	2.3566	2.0297	1.0986
153	2.5286	2.5286	3.0445	2.5286	3.0445	2.5286	2.3566
	2.3566	3.0445	3.0445	2.3566	2.3566	1.6687	1.3863

APPENDIX E

An Example of the Comparison Coefficient

Given two judges who cluster 20 observations (numbered 1 through 20) into groups as shown below:

Judge X			Judge Y		
Cluster	1	1,2,3,4,7	cluster	1	5,6,7
Cluster	2	5,6,11,12,13	cluster	2	1,2,3,4,9,15
Cluster	3	8,9,10,14,15, 16,17,18	cluster	3	8,11,12
Cluster	4	19,20	cluster	4	10,13,14,16,17, 18,19,20

The contingency table appears below with marginal (row) totals.

		Judge X				
		1	2	3	4	
Judge Y	1	1	2	0	0	3
	2	4	0	2	0	6
	3	0	2	1	0	3
	4	0	1	5	2	8
		5	5	8	2	20

Step 1: Find the sum of squares of the table entries.

$$1 + 4 + 16 + 4 + 4 + 1 + 1 + 25 + 4 = 60$$

Step 2: Find best possible sum of squares.
(Read in two columns)

	Judge X	Judge Y
	# obs in clusters	# obs in clusters
	3	5
	6	5
	3	8
	8	2
Max	8	8

Min of Max's = 8

Subtract Minimax from the max element and repeat

	3	5
	6	5
	3	0
	0	2
Max	6	5

Minimax = 5

Subtracting

	3	0
	1	5
	3	0
	0	2
Max	3	5

Minimax = 3

Subtracting

	0	0
	1	2
	3	0
	0	2
Max	3	2

Minimax = 2

Subtracting

	0	0
	1	0
	1	0
	0	2
Max	1	2

Minimax = 1

Subtracting

	0	0
	0	0
	1	0
	0	1
Max	1	1

Minimax = 1

Subtracting

0	0
0	0
0	0
0	0

Finished Step 2

Step 3: Sum the squares of Minimax's

$$54 + 25 + 9 + 4 + 1 + 1 = 104$$

Best possible sum of squares $= 104$

$$\text{Comparison coefficient} = \frac{\text{Actual sum of squares}}{\text{Best Possible Sum of Squares}}$$

$$= \frac{60}{104} = 0.58$$

APPENDIX F

```
//AIKSWKEE JOB (1642,0526,RJ74),'AIKEN SMC 2132',TIME=5
// EXEC FORTCLG,REGION.GO=200K
//FORT.SYSIN DD *
C       K-MEANS ITERATIVE CLUSTERING PROGRAM BY D.J. MCRAE
C
C       THIS PROGRAM ENABLES THE USER TO CLUSTER A DATA MATRIX
C       UP TO SIZE (600 X 20) INTO A MAXIMUM OF 20 CLUSTERS.
C       SEVERAL OPTIONS ARE AVAILABLE WITH RESPECT TO THE
C       METHOD OF COMPUTATION AND CRITERIA TO ACCOMPLISH THE
C       CLUSTERING.  THE PROGRAM MAY BE USED ON EITHER
C       THE CP/CMS TERMINALS OR BY THE CARD READER. THE APPROPRIATE
C       METHOD FOR EACH IS DESCRIBED BELOW.
C
C       CP/CMS TERMINAL USE
C
C               THE PROGRAM SHOULD BE READ INTO CP AS IS, BUT
C       PRECEDED BY THE FOLLOWING CARDS, STARTING IN COLUMN ONE
C
C               CP67USERID  XXXXG
C               OFFLINE READ RRREAD FORTRAN
C
C               THE DATA MATRIX CAN BE PLACED IN THE COMPUTER
C       EITHER BY USE OF THE OFFLINE READ OR BY TYPING IN THE
C       INFORMATION ON THE TERMINAL.  IF OFFLINE READ IS USED THE DATA
C       MUST BE IN THE SPECIFIED FORMAT PRECEDED BY THE FOLLOWING
C       TWO CARDS, STARTING IN COLUMN ONE:
C
C               CP67USERID  XXXXG
C               OFFLINE READ FILE FT04FYYY
C
C               THE PROGRAM CAN THEN BE EXECUTED BY FIRST ISSUING THE
C       CMS COMMAND
C                           F RRREAD
C       FOLLOWED BY
C                           $ RRREAD
C
C
C       OS/CARD READER USE
C
C               THE PROGRAM MUST CONTAIN THE FOLLOWING INFORMATION
C       IN THE FOLLOWING ORDER:
C                   STANDARD GREEN JOB CARD,TIME=(AS DESIRED)
C                   // EXEC FORTCLG,REGION.GO=200K
C                   //FORT.SYSIN DD *
C                       MAIN PROGRAM
C                   /*
C                   //GO.SYSIN DD *
C                       DATA DECK
```

```
C                 /*                                                    RRR00460
C                                                                       RRR00470
C                                                                       RRR00480
C      DATA  INPUT DECK                                                 RRR00490
C                                                                       RRR00500
C             THE FIRST CARD OF THE DATA DECK IS THE TITLE CARD.        RRR00510
C      IT MAY CONTAIN ANY ALPHA NUMERIC TITLE IN COLUMS 1 - 80.         RRR00520
C                                                                       RRR00530
C             THE SECOND CARD IS THE PROBLEM CARD.  IT CONTAINS         RRR00540
C      INTEGERS IN THE FIRST 13 COLLUMNS IN THE FOLLOWING MANNER:       RRR00550
C          COL 1-4:    NUMBER OF OBSERVATIONS                           RRR00560
C          COL 5-6:    NUMBER OF VARIABLES PER OBSERVATION              RRR00570
C          COL 7-8:    NUMBER OF CLUSTERS DESIRED                       RRR00580
C          COL 9:      CRITERION TO BE USED IN THE EVALUATION:          RRR00590
C                         1 = TRACE W                                   RRR00600
C                         2 = DETERMINANT W                             RRR00610
C                         3 = LARGEST ROOT OF W(INVERSE)*B              RRR00620
C                         4 = TRACE W(INVERSE)*B                        RRR00630
C          COL 10:     STANDARDIZATION PARAMETER:                       RRR00640
C                         0 = THE PROGRAM WILL NOT STANDARDIZE DATA     RRR00650
C                         1 = THE PROGRAM WILL STANDARDIZE DATA         RRR00660
C          COL 11:     DISTANCE PARAMETER                               RRR00670
C                         0 = EUCLIDIAN DISTANCE                        RRR00680
C                         1 = SCALED EUCLIDIAN DISTANCE                 RRR00690
C                         2 = MAHALANOBIS DISTANCE                      RRR00700
C          COL 12:     DATA PARAMETER                                   RRR00710
C                         0 = DATA IS TO BE READ IN                     RRR00720
C                         1 = NO DATA READ IN: PREVIOUS DATA IS         RRR00730
C                             TO BE REANALYZED USING CURRENT PROBLEM    RRR00740
C          COL 13:     TIMING PARAMETER                                 RRR00750
C                         0 = NO OBSERVATIONS CONSIDERED IN             RRR00760
C                             INDIVIDUAL SWITCHES                       RRR00770
C                         1 - 8 = AN INCREASING NUMBER OF OBSERVATICNS  RRR00780
C                         9 = ALL OBSERVATIONS USED                     RRR00790
C          NOTE: THE TIMING PARAMETER DETERMINED HCW EXTENSIVE THE      RRR00800
C                DOUBLE CHECK OF THE CLUSTER SOLUTION IS TO BE.         RRR00810
C                NORMALLY '9' IS USED UNLESS THE DATA MATRIX IS LARGE   RRR00820
C                                                                       RRR00830
C      THIS CARD MUST BE IN (I4,2I2,5I1) FORMAT                         RRR00840
C                                                                       RRR00850
C             THE THIRD CARD IS THE VARIABLE FORMAT CARD.  THIS CARD    RRR00860
C      MUST BE PRESENT UNLESS THE DATA PARAMETER IN THE PROBLEM         RRR00870
C      CARD IS '1'.  THE FORMAT STATEMENT IS THAT TYPE WHICH THE DATA   RRR00880
C      CARDS ARE IN.  FOR EXAMPLE IF THE DATA CARD HAS DATA:            RRR00890
C             ABCD    1.2    1.59                                       RRR00900
C      STARTING IN COL 1, THE VARIABLE FORMAT  CARE WOULD               RRR00910
C      CONTAIN THE FOLLOWING STARTING IN COL 1:                         RRR00920
C                (1A4,4X,F3.1,4X,F4.2)                                  RRR00930
```

```
C                                                                       RRR00940
C           THE NEXT CARDS ARE THE DATA CARDS.  THEY MAY CONTAIN        RRR00950
C     DATA IN COLUMNS 1 - 72.                                           RRR00960
C                                                                       RRR00970
C           FOR REANALYSIS OF THE SAME DATA,  ADD A NEW TITLE CARD     RRR00980
C     FOLLOWED BY A NEW PROBLEM CARD,WITH A '1' IN COLUMN 11            RRR00990
C     OF THE NEW PROBLEM CARD.                                          RRR01000
C           FOR ANALYSIS OF NEW DATA, SIMPLY PLACE THE NEW DATA         RRR01010
C     SET, INCLUDING NEW TITLE CARD, PROBLEM CARD, AND FORMAT CARD,     RRR01020
C     AFTER THE PREVIOUS SET                                            RRR01030
C                                                                       RRR01040
C     FOR THE PROGRAM TO EXIT NORMALLY, TWO BLANK CARDS MUST FOLLOW     RRR01050
C     THE LAST DATA BATCH.                                              RRR01060
C                                                                       RRR01070
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR01080
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR01090
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20), RRR01100
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20),NS(600),
     4TOP(20,20),BOT(20),UP(20,20),DOWN(20)
C                                                                       RRR01120
C                                                                       RRR01130
C     DESCRIPTION OF COMMON AREA:                                       RRR01140
C                                                                       RRR01150
C         NOBS = NUMBER OF OBSERVATIONS                                 RRR01160
C         NVARS = NUMBER OF VARIABLES                                   RRR01170
C         NGPS = NUMBER OF CLUSTERS                                     RRR01180
C         ICRIT = CRITERION TO BE OPTIMIZED (SEE ABOVE)                 RRR01190
C         NOSTAN = STANDARDIZATION PARAMETER (SEE ABOVE)                RRR01200
C         IDIST = DISTANCE PARAMETER  (SEE ABOVE)                       RRR01210
C         IFINE = ESCAPE PARAMETER:  IF IFINE IS SET EQUAL TO '1',      RRR01220
C                 SOMETHING IS WRONG AND THE APPROPRIATE ERROR          RRR01230
C                 MESSAGE IS PRINTED OUT; THE PROGRAM GOES ON TO THE    RRR01240
C                 NEXT PROBLEM                                          RRR01250
C          KTIME = TIMING PARAMETER (SEE ABOVE)                         RRR01260
C         IDENT = OBSERVATION IDENTIFICATIONS (A4 FORMAT):  READ FROM   RRR01270
C                 EACH DATA CARD                                        RRR01280
C         DATA = AREA IN WHICH THE DATA MATRIX IS STORED                RRR01290
C         T = CROSS-PRODUCTS MATRIX                                     RRR01300
C         B = BETWEEN-CLUSTERS MATRIX                                   RRR01310
C         W = WITHIN-CLUSTERS MATRIX                                    RRR01320
C         WFCT = CHOLESKY FACTOR OF THE WITHIN-CLUSTERS MATRIX          RRR01330
C         SVCEN = CLUSTER CENTERS (MEANS)                               RRR01340
C         IDATA = CLUSTER IDENTIFICATION FOR EACH OBSERVATION           RRR01350
C         NISV = CLUSTER SIZES (NUMBER OF OBSERVATIONS IN EACH CLUSTER) RRR01360
C         VMEAN = VARIABLE MEANS                                        RRR01370
C         SD = VARIABLE STANDARD DEVIATIONS                             RRR01380
C          XVEC = TEMPORARY STORAGE                                     RRR01390
C          YVEC = TEMPORARY STORAGE                                     RRR01400
```

```
C         BT = TEMPORARY STORAGE FOR BETWEEN-CLUSTERS MATRIX            RRR01410
C          NISVT, SVCENT, IDATAT:  TEMPORARY STORAGE SERVING THE        RRR01420
C               SAME FUNCTIONS AS NISV, SVCEN, AND IDATA                RRR01430
C          VEC = EIGENVECTORS                                           RRR01440
C         EIG = EIGENVALUES                                             RRR01450
C                                                                       RRR01460
C                                                                       RRR01470
      NNOBS=0
C     NNN IS USED 3 TIMES IN NEXT SEVERAL CARDS.
C     CHECK THIS CARD FOR CORRECT NOBS
      NNN=190
      READ(5,899)(NS(I),I=1,NNN)
899   FORMAT(25I3)
      WRITE(6,888)(NS(I),I=1,NNN)
888   FORMAT(' ',25I3)
      DO 200 I=1,NNN
200   NNOBS=NNOBS+NS(I)
      WRITE(6,887) NNOBS
887   FORMAT('0','TOTAL NUMBER OF OBSERVATIONS (WEIGHTED) IS ',I8)
100   CALL PRELIM(CRIT,NNOBS)
C     IFINE IS ESCAPE PARAMETER                                         RRR01490
      IF (IFINE.EQ.1) GO TO 400                                         RRR01500
      IF (IFINE.EQ.2) GO TO 500                                         RRR01510
      CALL RANDST (CRIT)                                                RRR01520
      IF (IFINE.EQ.1) GO TO 400                                         RRR01530
  300 CALL KMEANS (CRIT)                                                RRR01540
      IF (IFINE.EQ.1) GO TO 400                                         RRR01550
      CALL ISWTCH (CRIT)                                                RRR01560
      GO TO 100                                                         RRR01570
C     IFINE GOT SET TO 1:  IF DATA HAS BEEN READ IN, RESET DATA AND     RRR01580
C     GO TO NEXT PROBLEM                                                RRR01590
  400 DO 410 I=1,NOBS                                                   RRR01600
      DO 410 J=1,NVARS                                                  RRR01610
      IF (NOSTAN.EQ.1) DATA(I,J) = DATA(I,J) * SD(J)                    RRR01620
  410 DATA(I,J) = DATA(I,J) + VMEAN(J)                                  RRR01630
      GO TO 100                                                         RRR01640
C     ALL DONE:  WRITE OUT LAST MESSAGE AND EXIT                        RRR01650
  500 WRITE (6,900)                                                     RRR01660
  900 FORMAT ('0 END OF ANALYSES.')                                     RRR01670
      STOP                                                              RRR01680
      END                                                               RRR01690
      SUBROUTINE PRELIM(CRIT,NNOBS)
C                                                                       RRR01710
C     THIS SUBROUTINE MAKES THE PRELIMINARY CALCULATIONS.               RRR01720
C     IT INPUTS THE DATA, CALCULATES THE MEANS AND VARIANCES FOR EACH   RRR01730
C     VARIABLE, STANDARDIZES (CONVERTS TO Z-SCORES) EACH VARIABLE IF    RRR01740
C     REQUESTED, AND CALCULATES THE CROSS-PRODUCTS MATRIX               RRR01750
C                                                                       RRR01760
```

```
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600),  RRR01770
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20),  RRR01780
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20),  RRR01790
     3 NISVT(20),SVCENT(20,20),IDATAT(6CC),VEC(20,20),EIG(20),NS(6CC),
     4TOP(20,20),BOT(20),UP(20,20),DOWN(20)
      DIMENSION TITLE(20),IFMT(20),VAR(20)                                RRR01810
C                                                                         RRR01820
C     INPUT SECTION:  READ IN TITLE CARC, PROBLEM CARD, OPTIONAL CARDS,  RRR01830
C     FORMAT CARD, AND DATA CARDS                                         RRR01840
C     WRITE OUT SOLUTION SPECIFICATIONS                                   RRR01850
C                                                                         RRR01860
      IFINE = 0                                                           RRR01870
      READ (5,900) (TITLE(I),I=1,20)                                      RRR01880
      WRITE (6,903) (TITLE(I),I=1,20)                                     RRR01890
      READ (5,901) NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,NODATA,KTIME        RRR01900
      IF (NOBS.EQ.0) GO TO 820                                            RRR01910
      IF (NOBS.GT.600) GO TO 815                                          RRR01920
      IF (NVARS.GT.20) GO TO 815                                          RRR01930
      IF (NGPS.GT.20) GO TO 815                                           RRR01940
      IF (ICRIT.GT.4) GO TO 815                                           RRR01950
      IF (NOSTAN.GT.1) GO TO 815                                          RRR01960
      IF (IDIST.GT.2) GO TO 815                                           RRR01970
      IF (NVARS.EQ.1) ICRIT=1                                             RRR01980
      IF (NVARS.EQ.1) IDIST=0                                             RRR01990
    4 IF (NODATA.EQ.1) GO TO 5                                            RRR02000
      READ (5,902) (IFMT(I),I=1,18)                                       RRR02010
    5 WRITE (6,904) NOBS,NVARS,NGPS                                       RRR02020
      GC TO (10,20,30,40),ICRIT                                           RRR02030
   10 WRITE (6,920)                                                       RRR02040
      GO TO 50                                                            RRR02050
   20 WRITE (6,921)                                                       RRR02060
      GO TO 50                                                            RRR02070
   30 WRITE (6,922)                                                       RRR02080
      GC TO 50                                                            RRR02090
   40 WRITE (6,923)                                                       RRR02100
   50 CONTINUE                                                            RRR02110
      IF (NOSTAN) 60,60,70                                                RRR02120
   60 WRITE (6,924)                                                       RRR02130
      GC TO 80                                                            RRR02140
   70 WRITE (6,925)                                                       RRR02150
   80 CONTINUE                                                            RRR02160
      IF (IDIST) 85,85,90                                                 RRR02170
   85 WRITE (6,926)                                                       RRR02180
      GO TO 95                                                            RRR02190
   90 IF (IDIST.EQ.2) GO TO 92                                            RRR02200
      WRITE (6,936)                                                       RRR02210
      GO TO 95                                                            RRR02220
   92 WRITE (6,927)                                                       RRR02230
```

```
   55 CONTINUE                                                          RRR02240
      IF (NODATA.EQ.1) GO TO 101                                        RRR02250
      WRITE (6,905) (IFMT(I),I=1,18)                                    RRR02260
      DO 100 I = 1,NOBS                                                 RRR02270
      READ (5,IFMT) IDENT(I),(DATA(I,J),J=1,NVARS)                      RRR02280
  100 CONTINUE                                                          RRR02290
      NODATA = 1                                                        RRR02300
      GO TO 102                                                         RRR02310
  101 WRITE (6,937)                                                     RRR02320
  102 WRITE (6,943) KTIME                                               RRR02330
C                                                                       RRR02340
C     CALCULATE VARIABLE MEANS AND VARIANCES                            RRR02350
C     SUBTRACT OUT OVERALL MEAN                                         RRR02360
C                                                                       RRR02370
      DO 105 J=1,NVARS                                                  RRR02380
      VMEAN (J) = 0.0                                                   RRR02390
  105 VAR (J) = 0.0                                                     RRR02400
      DO 110 I = 1,NOBS                                                 RRR02410
      DO 110 J = 1,NVARS                                                RRR02420
110   VMEAN(J)=VMEAN(J)+NS(I)*DATA(I,J)/NNOBS
      DO 120 I=1,NOBS
      DO 120 J=1,NVARS                                                  RRR02450
      DATA(I,J) = DATA(I,J)-VMEAN(J)                                    RRR02460
120   VAR(J)=VAR(J)+NS(I)*(DATA(I,J)**2)/(NNOBS-1)
      DO 125 J = 1,NVARS                                                RRR02480
      IF (VAR(J).LE.0.000001) MFLAG=1                                   RRR02490
  125 SD(J) = SQRT (VAR(J))                                             RRR02500
      IF (MFLAG.EQ.1) GO TO 825                                         RRR02510
C                                                                       RRR02520
C    CALCULATE T = X'X, THE CROSS-PRODUCTS MATRIX                       RRR02530
C                                                                       RRR02540
  131 DO 135 K=1,NVARS                                                  RRR02550
      DO 135 J=K,NVARS                                                  RRR02560
  135 T(K,J) = 0.0                                                      RRR02570
      DO 140 K = 1,NVARS                                                RRR02580
      DO 140 J=K,NVARS                                                  RRR02590
      DO 140 I = 1,NOBS                                                 RRR02600
140   T(K,J)=T(K,J)+NS(I)*DATA(I,K)*DATA(I,J)
C                                                                       RRR02620
C    OUTPUT SECTION: WRITE OUT MEANS, STANDARD DEVIATIONS, AND CROSS-   RRR02630
C                    PRODUCTS MATRIX T                                  RRR02640
C                                                                       RRR02650
      WRITE (6,908)                                                     RRR02660
      WRITE (6,907) (VMEAN(J),J=1,NVARS)                                RRR02670
      WRITE (6,909)                                                     RRR02680
      WRITE (6,907) (SD(J),J=1,NVARS)                                   RRR02690
      WRITE (6,911)                                                     RRR02700
      DO 810 I = 1,NVARS                                                RRR02710
```

```
      WRITE (6,907) (T(J,I),J=1,I)                                      RRR02720
  810 CONTINUE                                                          RRR02730
      IF (NOSTAN.EQ.0) GO TO 180                                        RRR02740
C                                                                       RRR02750
C     STANDARDIZE IF REQUESTED                                          RRR02760
C                                                                       RRR02770
      DO 160 J=1,NVARS                                                  RRR02780
      DO 150 I=1,NOBS                                                   RRR02790
  150 DATA(I,J) = DATA(I,J) / SD(J)                                     RRR02800
      DO 160 K=J,NVARS                                                  RRR02810
  160 T(J,K) = (1.0/SD(J))*T(J,K)*(1.0/SD(K))                           RRR02820
  180 RETURN                                                            RRR02830
  815 IFINE = 1                                                         RRR02840
      IF (NODATA.EQ.0) IFINE=2                                          RRR02850
      WRITE (6,935)                                                     RRR02860
      RETURN                                                            RRR02870
  820 IFINE = 2                                                         RRR02880
      RETURN                                                            RRR02890
  825 WRITE (6,945) J                                                   RRR02900
      IFINE = 1                                                         RRR02910
      RETURN                                                            RRR02920
C                                                                       RRR02930
C     FORMAT STATEMENTS                                                 RRR02940
C                                                                       RRR02950
  900 FORMAT (20A4)                                                     RRR02960
  901 FORMAT (I4,2I2,5I1)                                               RRR02970
  902 FORMAT (18A4)                                                     RRR02980
  903 FORMAT ('1',20A4)                                                 RRR02990
  904 FORMAT ('0 THE NUMBER OF OBSERVATIONS IS ',I4,/,'  THE NUMBER OF VRRR03000
     1ARIABLES IS ',I2,/,'  THE NUMBER OF GROUPS (INITIAL) IS ',I2)     RRR03010
  905 FORMAT ('0 THE INPUT FORMAT IS ',18A4)                            RRR03020
  907 FORMAT (' ',10(F11.3,1X))                                         RRR03030
  908 FORMAT ('0THE VARIABLE MEANS ARE')                                RRR03040
  909 FORMAT ('0THE VARIABLE STANDARD DEVIATIONS ARE')                  RRR03050
  911 FORMAT ('0THE CROSS-PRODUCTS MATRIX IS ')                         RRR03060
  920 FORMAT ('0 THE CRITERION IS TO MINIMIZE TRACE W')                 RRR03070
  921 FORMAT ('0 THE CRITERION IS TO MINIMIZE DETERMINANT W')           RRR03080
  922 FORMAT ('0 THE CRITERION IS TO MAXIMIZE THE LARGEST ROOT OF W-1B')RRR03090
  923 FORMAT ('0 THE CRITERION IS TO MINIMIZE THE TRACE OF W-1B')       RRR03100
  924 FORMAT ('0 THE VARIABLES WILL NOT BE STANDARDIZED')               RRR03110
  925 FORMAT ('0 THE VARIABLES WILL BE STANDARDIZED')                   RRR03120
  926 FORMAT ('0 EUCLIDIAN DISTANCE WILL BE USED')                      RRR03130
  927 FORMAT ('0 MAHALANOBIS DISTANCE WILL BE USED')                    RRR03140
  935 FORMAT ('1 ERROR IN PROBLEM CARD.  SKIPPING PROBLEM.')            RRR03150
  936 FORMAT ('0 WEIGHTED EUCLIDIAN DISTANCE WILL BE USED.')            RRR03160
  937 FORMAT ('0PREVIOUS DATA BEING USED ')                             RRR03170
  943 FORMAT ('0THE TIMING PARAMETER FOR INDIVIDUAL SWITCHES IS ',I1)   RRR03180
  945 FORMAT ('0 WARNING:  VARIABLE ',I2,' HAS ZERO VARIANCE.')         RRR03190
```

```
      END                                                               RRR03200
      SUBROUTINE RANDST (CRIT)                                          RRR03210
C                                                                       RRR03220
C     THIS SUBROUTINE DETERMINES THE INITIAL CLUSTER CONFIGURATION.     RRR03230
C                                                                       RRR03240
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR03250
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR03260
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20), RRR03270
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20),NS(600),
     4TOP(20,20),BOT(20),UP(20,20),DOWN(20)
      DIMENSION ISTRT(20)                                               RRR03290
C                                                                       RRR03300
C     ONE PASS K-MEANS TO DETERMINE INITIAL CLUSTER SOLUTION            RRR03310
C                                                                       RRR03320
C     RANDOM NUMBER STARTER                                             RRR03330
  200 IX=19739                                                          RRR03340
      NGPST=NGPS                                                        RRR03350
C     DO 260: GETTING THREE INITIAL CONFIGURATIONS; THE ONE WITH THE    RRR03360
C     BEST CRITERION VALUE WILL BE USED                                 RRR03370
      DO 260 IBSRT = 1,3                                                RRR03380
      DO 205 M = 1,NGPS                                                 RRR03390
      BOT(M)=0.0
      DO 202 KKK=1,NVARS
  202 TOP(M,KKK)=0.0
      NISVT(M)=1                                                        RRR03400
C     FIND A RANDOM OBSERVATION TO SERVE AS INITIAL CLUSTER CENTER      RRR03410
C     FOR EACH CLUSTER                                                  RRR03420
  201 RAND = URAND(IX)                                                  RRR03430
      ISTRT(M) = RAND * NOBS                                            RRR03440
      IF (ISTRT(M).EQ.0) ISTRT(M)=1                                     RRR03450
      IF (M.EQ.1) GO TO 203                                             RRR03460
      MM1 = M-1                                                         RRR03470
      DO 204 MM=1,MM1                                                   RRR03480
      IF (ISTRT(M).EQ.ISTRT(MM)) GO TO 201                              RRR03490
  204 CONTINUE                                                          RRR03500
  203 I = ISTRT(M)                                                      RRR03510
      BOT(M)=NS(I)
      DO 205 J=1,NVARS                                                  RRR03520
      TOP(M,J)=DATA(I,J)
  205 SVCENT(M,J) = DATA(I,J)                                           RRR03530
      ITEMP = IDIST                                                     RRR03540
      IDIST = 0                                                         RRR03550
C     DO FOR EACH OBSERVATION                                           RRR03560
      DO 225 I=1,NOBS                                                   RRR03570
C     FIND EUCLIDIAN DISTANCE TO EACH CLUSTER CENTER                    RRR03580
      DO 215 M=1,NGPS                                                   RRR03590
      DO 210 J=1,NVARS                                                  RRR03600
      XVEC(J)=DATA(I,J)                                                 RRR03610
```

```
  210 YVEC(J) = SVCENT(M,J)                                               RRR03620
      CALL DISTCE (XVEC,YVEC,DISTA)                                       RRR03630
      IF (IFINE.EQ.1) RETURN                                              RRR03640
      IF (M.EQ.1) GO TO 212                                               RRR03650
      IF (DISTA.GE.SMDIST) GO TO 215                                      RRR03660
  212 SMDIST = DISTA                                                      RRR03670
      IDATAT(I) = M                                                       RRR03680
  215 CONTINUE                                                            RRR03690
C     ASSIGN OBSERVATION TO CLOSEST CLUSTER AND UPDATE THAT CLUSTER       RRR03700
C     CENTER                                                              RRR03710
      K = IDATAT(I)                                                       RRR03720
      NISVT(K) = NISVT(K) + 1                                             RRR03730
      BOT(K)=BOT(K)+NS(I)
      DO 225 J = 1,NVARS                                                  RRR03740
      TOP(K,J)=TOP(K,J)+NS(I)*DATA(I,J)
225   SVCENT(K,J)=TOP(K,J)/BOT(K)
      IDIST = ITEMP                                                       RRR03760
C     END OF DO FOR EACH OBSERVATION                                      RRR03770
C     RECALCULATE CLUSTER CENTERS TO ELIMINATE INITIAL RANDOM OBSERVA-    RRR03780
C     TIONS                                                               RRR03790
      DO 230 M=1,NGPS                                                     RRR03800
      BOT(M)=0.0
      NISVT(M) = NISVT(M)-1                                               RRR03810
      DO 230 J=1,NVARS                                                    RRR03820
      TOP(M,J)=0.0
  230 SVCENT(M,J)=0.                                                      RRR03830
      DO 235 I=1,NOBS                                                     RRR03840
      M = IDATAT(I)                                                       RRR03850
      BOT(M)=BOT(M)+NS(I)
      DO 235 J=1,NVARS                                                    RRR03860
      TOP(M,J)=TOP(M,J)+NS(I)*DATA(I,J)
235   SVCENT(M,J)=TOP(M,J)/BOT(M)
C                                                                         RRR03880
C     CALCULATE THE B AND W MATRICES                                      RRR03890
C     CALCULATE THE CRITERION VALUE                                       RRR03900
C                                                                         RRR03910
      IDIR=2                                                              RRR03920
      IF (ICRIT.EQ.1) IDIR=1                                              RRR03930
      CALL WCALC (SVCENT,NISVT,NGPST,IDIR)                                RRR03940
      IF (IFINE.EQ.1) RETURN                                              RRR03950
      CALL CRITON (CRIT)                                                  RRR03960
      IF (IFINE.EQ.1) RETURN                                              RRR03970
C     WHICH INITIAL CONFIGURATION                                         RRR03980
      IF (IBSRT.GT.1) GO TO 250                                           RRR03990
C     FIRST INITIAL CONFIGURATION:  BCRIT IS THE BEST CRITERION           RRR04000
  245 BCRIT = CRIT                                                        RRR04010
C     SAVE CLUSTER SIZES (NISV), CLUSTER CENTERS (SVCEN), CLUSTER         RRR04020
C     LISTS (LSTSV), AND OBSERVATION ID'S (IDATA)                         RRR04030
```

```
      DO 247 M = 1,NGPS                                                 RRR04040
      NISV(M) = NISVT(M)                                                RRR04050
      DOWN(M)=BOT(M)
      DO 247 L = 1,NVARS                                                RRR04060
      UP(M,L)=TOP(M,L)
  247 SVCEN(M,L) = SVCENT(M,L)                                          RRR04070
      DO 249 I=1,NOBS                                                   RRR04080
  249 IDATA(I) = IDATAT(I)                                              RRR04090
      GO TO 260                                                         RRR04100
C     SECOND OR THIRD INITIAL CONFIGURATION                             RRR04110
  250 IF (CRIT.GE.BCRIT) GO TO 260                                      RRR04120
  255 BCRIT = CRIT                                                      RRR04130
      DO 257 M = 1,NGPS                                                 RRR04140
      NISV(M) = NISVT(M)                                                RRR04150
      DOWN(M)=BOT(M)
      DO 257 L = 1,NVARS                                                RRR04160
      UP(M,L)=TOP(M,L)
  257 SVCEN(M,L) = SVCENT(M,L)                                          RRR04170
      DO 259 I=1,NOBS                                                   RRR04180
  259 IDATA(I) = IDATAT(I)                                              RRR04190
  260 CONTINUE                                                          RRR04200
C     ALL DONE                                                          RRR04210
      RETURN                                                            RRR04220
      END                                                               RRR04230
      SUBROUTINE KMEANS (CRIT)                                          RRR04240
C                                                                       RRR04250
C          K-MEANS.  EACH OBSERVATION IS ASSIGNED TO THE CLOSEST CLUSTER RRR04260
C          CENTER.                                                      RRR04270
C                                                                       RRR04280
C                                                                       RRR04290
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR04300
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR04310
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20), RRR04320
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20),NS(600),
     4TOP(20,20),BOT(20),UP(20,20),DOWN(20)
C     INITIALIZING FLAGS                                                RRR04340
C     AND TEMPORARY STORAGE FOR NISV, SVCEN                             RRR04350
  200 NGPST = NGPS                                                      RRR04360
      JFLAG = 0                                                         RRR04370
      BCRIT = CRIT                                                      RRR04380
      DO 201 M=1,NGPS                                                   RRR04390
      NISVT(M) = NISV(M)                                                RRR04400
      BOT(M)=DOWN(M)
      DO 201 J=1,NVARS                                                  RRR04410
      TOP(M,J)=UP(M,J)
  201 SVCENT(M,J) = SVCEN(M,J)                                          RRR04420
C     RECALCULATING WITHIN-CLUSTERS MATRIX                              RRR04430
      IDIR=1                                                            RRR04440
```

```
      IF (IDIST.EC.2) IDIR=2                                            RRRC4450
      CALL WCALC (SVCENT,NISVT,NGPST,IDIR)                              RRR04460
      IF (IFINE.EQ.1) RETURN                                            RRR04470
C     INITIALIZING TEMPORARY STORAGE FOR LSTSV, IDATA                   RRR04480
      DO 210 I = 1,NOBS                                                 RRR04490
  210 ICATAT(I) = IDATA(I)                                              RRR04500
C     MAJOR DO LOOP:  DO FOR EACH OBSERVATION                           RRR04510
      DO 400 I = 1,NOBS                                                 RRR04520
C                                                                       RRR04530
C      BEGIN K-MEANS SECTION                                            RRR04540
C                                                                       RRR04550
C     CALCULATE VECTOR OF DISTANCES FROM OBSERVATION TO EACH CLUSTER    RRR0456C
C     CENTER                                                            RRR04570
  325 DC 335 M=1,NGPST                                                  RRR04580
      DO 330 J=1,NVARS                                                  RRR04590
      XVEC (J) = DATA (I,J)                                             RRR04600
  330 YVEC (J) = SVCENT (M,J)                                           RRR0461C
      CALL DISTCE (XVEC,YVEC,DISTB)                                     RRR04620
      IF (IFINE.EQ.1) RETURN                                            RRR04630
      IF (M.EQ.1) GO TO 332                                             RRR04640
      IF (DISTB.GE.SMDIST) GO TO 335                                    RRR04650
  332 SMDIST = DISTB                                                    RRR04660
      IDGP = M                                                          RRR04670
  335 CONTINUE                                                          RRR04680
C     IS OBSERVATIGN ALREADY IN THE CLOSEST CLUSTER? IF YES, SKIP THIS  RRR04690
C     SECTION                                                           RRR04700
      IF (IDGP.EQ.IDATAT(I)) GO TO 360                                  RRR04710
C     IOLD IS OLD CLUSTER ASSIGNMENT                                    RRR04720
  346 IOLD = IDATAT(I)                                                  RRR04730
C     INEW IS NEW CLUSTER ASSIGNMENT                                    RRR04740
      INEW = IDGP                                                       RRR04750
      JFLAG = 1                                                         RRR04760
C     RECALCULATE CLUSTER CENTERS                                       RRR04770
      BOT(IOLD)=BOT(IOLD)-NS(I)
      BOT(INEW)=BOT(INEW)+NS(I)
      DC 348 J = 1,NVARS                                                RRR04780
      TOP(IOLD,J)=TCP(IOLD,J)-NS(I)*DATA(I,J)
      SVCENT(IOLD,J)=TOP(IOLD,J)/BOT(ICLD)
      TOP(INEW,J)=TOP(INEW,J)+NS(I)*DATA(I,J)
  348 SVCENT(INEW,J)=TOP(INEW,J)/BOT(INEW)
C     ADJUST NISV                                                       RRR04830
      NISVT(IOLD) = NISVT(IOLD)-1                                       RRR04840
      NISVT(INEW) = NISVT(INEW)+1                                       RRR04850
C     ADJUST IDATA                                                      RRR04860
      IDATAT(I) = IDGP                                                  RRRC4870
C     RECALCULATE WITHIN-CLUSTERS MATRIX FOR USE IN COMPUTING SCALED    RRR04880
C     EUCLIDIAN AND MAHALANOBIS DISTANCE                                RRR04890
      IF (IDIST.EQ.0) GO TO 360                                         RRR04900
```

```
      CALL WCALC (SVCENT,NISVT,NGPST,IDIR)                            RRR04910
      IF (IFINE.EQ.1) RETURN                                          RRR04920
  360 CONTINUE                                                        RRR04930
C     DONE WITH MAJOR DO LOOP                                         RRR04940
  400 CONTINUE                                                        RRR04950
C                                                                     RRR04960
C      CALCULATE THE CRITERION                                        RRR04970
C                                                                     RRR04980
C     RECALCULATE SVCEN:  ACCURACY MEASURE                            RRR04990
      DO 401 M=1,NGPST                                                RRR05000
      BOT(M)=0.0
      DO 401 J=1,NVARS                                                RRR05010
      TOP(M,J)=0.0
  401 SVCENT(M,J) = 0.                                                RRR05020
      DO 402 I=1,NOBS                                                 RRR05030
      M = IDATAT(I)                                                   RRR05040
      BOT(M)=BOT(M)+NS(I)
      DO 402 J=1,NVARS                                                RRR05050
      TOP(M,J)=TOP(M,J)+NS(I)*DATA(I,J)
402   SVCENT(M,J)=TOP(M,J)/BOT(M)
C     RECALCULATE WITHIN-CLUSTERS MATRIX AND CRITERION VALUE BASED ON RRR05070
C     NEW CLUSTER CENTER VALUES                                       RRR05080
      IDIR = 2                                                        RRR05090
      IF (ICRIT.EQ.1) IDIR = 1                                        RRR05100
      CALL WCALC (SVCENT,NISVT,NGPST,IDIR)                            RRR05110
      IF (IFINE.EQ.1) RETURN                                          RRR05120
  405 CALL CRITON (CRIT)                                              RRR05130
      IF (IFINE.EQ.1) RETURN                                          RRR05140
C     IF CRITERION BETTER THAN BEFORE?  IF YES, THEN ANOTHER ITERATION; RRR05150
C     IF NO, FINISH KMEANS                                            RRR05160
      IF (CRIT.GE.BCRIT) GO TO 535                                    RRR05170
C                                                                     RRR05180
C      ANOTHER ITERATION                                              RRR05190
C                                                                     RRR05200
C     PUT TEMPORARY VALUES INTO PERMANENT LOCATIONS                   RRR05210
  515 NGPS=NGPST                                                      RRR05220
      DO 520 M = 1,NGPS                                               RRR05230
      DOWN(M)=BOT(M)
      NISV(M) = NISVT (M)                                             RRR05240
      DO 520 J = 1,NVARS                                              RRR05250
      UP(M,J)=TOP(M,J)
  520 SVCEN (M,J) = SVCENT (M,J)                                      RRR05260
      DO 530 I = 1,NOBS                                               RRR05270
  530 IDATA(I) = IDATAT(I)                                            RRR05280
      GO TO 200                                                       RRR05290
C                                                                     RRR05300
C      FINISH                                                         RRR05310
C                                                                     RRR05320
```

```
C     JFLAG = 0 MEANS NO CHANGES HAVE BEEN MADE DURING THE LAST          RRR05330
C     ITERATION:  ITERATIONS CONVERGED                                   RRR05340
  535 IF (JFLAG.EQ.0) RETURN                                             RRR05350
C     JFLAG = 1 MEANS CHANGES HAVE BEEN MADE BUT THE CRITERION VALUE     RRR05360
C     GOT WORSE:  ITERATIONS NOT CONVERGING                              RRR05370
  540 WRITE (6,940)                                                      RRR05380
      WRITE (6,942) CRIT                                                 RRR05390
      WRITE (6,943) BCRIT                                                RRR05400
C     RECALCULATE WITHIN-CLUSTERS MATRIX AND RESET CRIT                  RRR05410
      IDIR=2                                                             RRR05420
      IF (ICRIT.EQ.1) IDIR=1                                             RRR05430
      CALL WCALC (SVCEN,NISV,NGPS,IDIR)                                  RRR05440
      IF (IFINE.EQ.1) RETURN                                             RRR05450
      CRIT=BCRIT                                                         RRR05460
      RETURN                                                             RRR05470
  940 FORMAT ('0 ITERATIONS NOT CONVERGING')                             RRR05480
  942 FORMAT ('0 THE CRITERION VALUE IS ',E12.6)                         RRR05490
  943 FORMAT ('0 THE BEST CRITERION VALUE IS ',E12.6)                    RRR05500
      END                                                                RRR05510
      SUBROUTINE ISWTCH (CRIT)                                           RRR05520
C                                                                        RRR05530
C     THIS SUBROUTINE CONSIDERS SWITCHING EACH OBSERVATION TO A          RRR05540
C     DIFFERENT CLUSTER.  THE SWITCH IS MADE IFF A BETTER CRITERION      RRR05550
C     VALUE RESULTS.                                                     RRR05560
C     THIS SUBROUTINE ALSO DEPENDS ON THE PARAMETER KTIME:  IT DETERMINESRRR05570
C     WHICH OBSERVATIONS ARE TO BE CONSIDERED.  FOR COMPLETE EXPLANATION RRR05580
C     OF THE KTIME PARAMETER, SEE THE PROGRAM DESCRIPTION.               RRR05590
C                                                                        RRR05600
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600),  RRR05610
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20),  RRR05620
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20),  RRR05630
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20),NS(600),
     4TOP(20,20),BOT(20),UP(20,20),DOWN(20)
      DIMENSION JFLAG (20)                                               RRR05650
C     SET UP THE IDIR FLAG                                               RRR05660
      IDIR=2                                                             RRR05670
      IF (ICRIT.EQ.1) IDIR=1                                             RRR05680
C     KTIME = 0 MEANS SKIP THIS HEURISTIC                                RRR05690
      IF (KTIME.EQ.0) GO TO 750                                          RRR05700
C     SET UP TEMPORARY STORAGE AREAS FOR NGPS, NISV, AND SVCEN           RRR05710
      NGPST=NGPS                                                         RRR05720
      DO 500 M = 1,NGPS                                                  RRR05730
      BOT(M)=DOWN(M)
      NISVT(M)=NISV(M)                                                   RRR05740
      JFLAG(M)=0                                                         RRR05750
      DO 500 J = 1,NVARS                                                 RRR05760
      TOP(M,J)=UP(M,J)
  500 SVCENT(M,J)=SVCEN(M,J)                                             RRR05770
```

```
C     SET IFLAG = 0:  WILL BE SET = 1 IF ANY SWITCH IS MADE
  60C IFLAG=0
C     MAJOR DO LOOP:  DO 700
      DO 700 M=1,NGPS
C     JFLAG (M) INDICATES WHETHER CLUSTER M HAS BEEN ALTERED
      IF (JFLAG(M).EQ.2) GO TO 700
C     KTIME = 9 MEANS ALL OBSERVATIONS WILL BE CONSIDERED
      IF (KTIME.EQ.9) GO TO 617
C     COMPUTE ALL DISTANCES FROM OBSERVATIONS TO CLUSTER CENTERS FOR THE
C     OBSERVATIONS IN CLUSTER M
      DO 605 J=1,NVARS
  605 YVEC(J) = SVCEN(M,J)
      BIGDIS = 0.
      DO 615 I=1,NOBS
      IF (IDATA(I).NE.M) GO TO 615
      DO 610 J=1,NVARS
  610 XVEC(J) = DATA(I,J)
      CALL DISTCE (XVEC,YVEC,DISTA)
C     LOCATE OBSERVATION WITH BIGGEST DISTANCE
      IF (BIGDIS.GE.DISTA) GO TO 615
      BIGDIS = DISTA
  615 CONTINUE
C     DIVIDE BIGDIS BY TIMING PARAMETER KTIME
      TIME = KTIME
      PSME = BIGDIS / TIME
  617 MOLD = M
C     DO FOR ALL OBSERVATIONS IN CLUSTER M
      DO 691 I=1,NOBS
      IF (IDATA(I).NE.M) GO TO 691
      KFLAG = 0
C     KTIME = 9 MEANS CONSICER ALL OBSERVATIONS
      IF (KTIME.EQ.9) GO TO 618
C     DO NO  CONSIDER SWITCHING THE OBSERVATION IF THE DISTANCE TO ITS
C     CURRENT CLUSTER CENTER IS LT BIGDIS / KTIME
      DO 619 J=1,NVARS
  619 XVEC(J) = DATA(I,J)
      CALL DISTCE (XVEC,YVEC,DISTA)
      IF (DISTA.LT.PSME) GO TO 691
  618 CONTINUE
      DO 690 MNEW=1,NGPS
      IF (NISV(MOLD).EQ.1) GO TO 690
      IF (MNEW.EQ.MOLD) GO TO 690
      IF (KFLAG.EQ.1) GO TO 690
C     COMPUTE NEW SVCEN BASED ON OBSERVATION BEING SWITCHED
      BOT(MNEW)=BCT(MNEW)+NS(I)
      BOT(MOLD)=BOT(MOLD)-NS(I)
      DO 620 J=1,NVARS
      TOP(MNEW,J)=TOP(MNEW,J)+NS(I)*DATA(I,J)
```

```
      SVCENT(MNEW,J)=TOP(MNEW,J)/BOT(MNEW)
      TCP(MOLD,J)=TOP(MOLD,J)-NS(I)*DATA(I,J)
      SVCENT(MOLD,J)=TOP(MOLD,J)/BOT(MOLD)
  620 CCNTINUE                                                          RRR06270
C    ADJUST NISV                                                        RRR06280
      NISVT(MNEW) = NISV(MNEW)+1                                        RRR06290
      NISVT(MOLD) = NISV(MOLD)-1                                        RRR06300
C    COMPUTE WITHIN-CLUSTERS MATRIX AND NEW CRITERICN VALUE             RRR06310
      CALL WCALC (SVCENT,NISVT,NGPST,IDIR)                              RRR06320
      IF (IFINE.EC.1) RETURN                                            RRR06330
      CALL CRITON (TCRIT)                                               RRR06340
      IF (IFINE.EQ.1) RETURN                                            RRR06350
C    MAKE THE SWITCH IFF NEW CRIT IS BETTER                             RRR06360
      IF (TCRIT.GE.CRIT) GO TO 680                                      RRR06370
C                                                                       RRR06380
C     MAKE THE SWITCH                                                   RRR06390
C                                                                       RRR06400
C    RESET FLAGS                                                        RRR06410
      JFLAG(M) = 1                                                      RRR06420
      JFLAG(MNEW) = 1                                                   RRR06430
      IFLAG=1                                                           RRR06440
      KFLAG = 1                                                         RRR06450
C    ADJUST CRIT, IDATA, NISV, SVCEN                                    RRR06460
      CRIT=TCRIT                                                        RRR06470
      ICATA(I) = MNEW                                                   RRR06480
      DCWN(MNEW)=BOT(MNEW)
      DCWN(MOLD)=BOT(MOLD)
      NISV(MNEW) = NISVT(MNEW)                                          RRR06490
      NISV(MOLD) = NISVT(MOLD)                                          RRR06500
  629 DO 630 J=1,NVARS                                                  RRR06510
      UP(MNEW,J)=TOP(MNEW,J)
      UP(MOLD,J)=TCP(MOLD,J)
      SVCEN(MNEW,J) = SVCENT(MNEW,J)                                    RRR06520
  630 SVCEN(MOLD,J) = SVCENT(MOLD,J)                                    RRR06530
      GO TO 690                                                         RRR06540
C    SWITCH WAS NOT MADE; RESET NISVT, SVCENT, TO VALUES PRESENT        RRR06550
C    BEFORE THE SWITCH WAS CONSIDERED                                   RRR06560
  680 NISVT(MNEW) = NISV(MNEW)                                          RRR06570
      NISVT(MOLD) = NISV(MOLD)                                          RRR06580
      BOT(MNEW)=DCWN(MNEW)
      BOT(MOLD)=DOWN(MOLD)
      DO 685 J=1,NVARS                                                  RRR06590
      TOP(MNEW,J)=UP(MNEW,J)
      TOP(MOLD,J)=UP(MOLD,J)
      SVCENT(MNEW,J) = SVCEN(MNEW,J)                                    RRR06600
  685 SVCENT(MOLD,J) = SVCEN(MOLD,J)                                    RRR06610
  690 CONTINUE                                                          RRR06620
  691 CONTINUE                                                          RRR06630
```

```
C     FINISH WITH CLUSTER M:  IF NO SWITCHES HAVE BEEN MADE, SET         RRR06640
C     JFLAG(M) = 2 AND GO TO NEXT CLUSTER;  IF SWITCHES HAVE BEEN MADE,  RRR06650
C     ADJLST LSTSV AND SET JFLAG(M) = 0 AND GO TO NEXT CLUSTER           RRR06660
  695 IF (JFLAG(M).EQ.0) GO TO 699                                        RRR06670
      JFLAG(M) = 0                                                        RRR06680
      GO TO 700                                                           RRR06690
  699 JFLAG(M) = 2                                                        RRR06700
  700 CONTINUE                                                            RRR06710
C     DONE WITH ALL CLUSTERS:  IFLAG = 1 MEANS SOME SWITCHES HAVE BEEN   RRR06720
C     MACE; GO BACK AND ITERATE                                          RRR06730
      IF (IFLAG.EQ.1) GO TO 600                                           RRR06740
C                                                                         RRR06750
C       ALL DONE; ACCURATELY CALCULATE MEANS AND CRITERION AND OUTPUT    RRR06760
C       THE RESULTS                                                      RRR06770
C                                                                         RRR06780
  750 WRITE (6,900)                                                       RRR06790
  900 FORMAT ('0 THE FINAL CLUSTER SOLUTION IS ')                         RRR06800
C     RECALCULATE CLUSTER CENTERS, WITHIN-CLUSTERS MATRIX, AND CRITERION RRR06810
C     VALUE                                                              RRR06820
      DO 715 M=1,NGPS                                                     RRR06830
      BCT(M)=0.0
      DO 715 J=1,NVARS                                                    RRR06840
      TCP(M,J)=0.0
  715 SVCEN(M,J) = 0.                                                     RRR06850
      DO 720 I=1,NOBS                                                     RRR06860
      M = IDATA(I)                                                        RRR06870
      BCT(M)=BCT(M)+NS(I)
      DO 720 J=1,NVARS                                                    RRR06880
      TOP(M,J)=TOP(M,J)+NS(I)*DATA(I,J)
720   SVCEN(M,J)=TOP(M,J)/BCT(M)
      CALL WCALC (SVCEN,NISV,NGPS,IDIR)                                   RRR06900
      IF (IFINE.EC.1) RETURN                                              RRR06910
      CALL CRITON (CRIT)                                                  RRR06920
      IF (IFINE.EQ.1) RETURN                                              RRR06930
C     CALL THE OUTPUT ROUTINE                                            RRR06940
      CALL OUTPUT (CRIT)                                                  RRR06950
      WRITE (6,901)                                                       RRR06960
  901 FORMAT ('0 END OF CLUSTER PROBLEM')                                 RRR06970
      RETURN                                                              RRR06980
      END                                                                 RRR06990
      SUBROUTINE CUTPUT (CRIT)                                            RRR07000
C                                                                         RRR07010
C       THIS SUBROUTINE PRINTS OUT THE CLUSTER SOLUTION                  RRR07020
C            FOR EACH CLUSTER - THE CLUSTER SIZE                         RRR07030
C                               THE CLUSTER CENTROID                     RRR07040
C                               THE OBSERVATIONS BELONGING TO THE CLUSTER RRR07050
C            THE WITHIN CELLS MATRIX                                     RRR07060
C            THE CRITERION VALUE                                         RRR07070
```

```
C                                                                       RRR07080
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR07090
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR07100
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20), RRR07110
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20),NS(600),
     4TOP(20,20),BOT(20),UP(20,20),DOWN(20)
C     UNSTANDARDIZE IF NECESSARY                                        RRR07130
      IF (NOSTAN.EQ.0) GO TO 40                                         RRR07140
      DO 20 M=1,NGPS                                                    RRR07150
      DO 20 J=1,NVARS                                                   RRR07160
   20 SVCEN(M,J) = SVCEN(M,J)*SD(J)                                     RRR07170
      DO 30 J=1,NVARS                                                   RRR07180
      DO 30 K=J,NVARS                                                   RRR07190
   30 T(J,K) = SD(J)*T(J,K)*SD(K)                                       RRR07200
      IDIR = 2                                                          RRR07210
      IF (ICRIT.EQ.1) IDIR=1                                            RRR07220
      CALL WCALC (SVCEN,NISV,NGPS,IDIR)                                 RRR07230
      IF (IFINE.EQ.1) RETURN                                            RRR07240
      CALL CRITON (CRIT)                                                RRR07250
      IF (IFINE.EQ.1) RETURN                                            RRR07260
C    ADD THE OVERALL MEAN BACK INTO EACH OBSERVATION: UNSTANDARDIZE     RRR07270
C     THE DATA IF NECESSARY                                             RRR07280
   40 DO 80 J=1,NVARS                                                   RRR07290
      DO 50 I=1,NOBS                                                    RRR07300
      IF (NOSTAN.EQ.1) DATA(I,J) = DATA(I,J)*SD(J)                      RRR07310
   50 DATA(I,J) = DATA(I,J) + VMEAN(J)                                  RRR07320
      DO 60 M=1,NGPS                                                    RRR07330
   60 SVCEN(M,J) = SVCEN(M,J) + VMEAN(J)                                RRR07340
   80 CONTINUE                                                          RRR07350
C    WRITE OUT NISV, SVCEN                                              RRR07360
   95 DO 115 M=1,NGPS                                                   RRR07370
      WRITE (6,900) M                                                   RRR07380
      WRITE (6,901) NISV(M)                                             RRR07390
      WRITE (6,902) (SVCEN(M,J),J=1,NVARS)                              RRR07400
      WRITE(7,915)(SVCEN(M,J),J=1,NVARS)
915   FORMAT(8F8.4,/8F8.4)
      WRITE (6,903)                                                     RRR07410
      LSTNO = NISV(M)                                                   RRR07420
      K = 0                                                             RRR07430
      DO 100 I=1,NOBS                                                   RRR07440
      IF (IDATA(I).NE.M) GO TO 100                                      RRR07450
      K = K+1                                                           RRR07460
      IDATAT(K) = IDENT(I)                                              RRR07470
  100 CONTINUE                                                          RRR07480
C     SORT THE OBSERVATION IDENTIFICATIONS                              RRR07490
      L=0                                                               RRR07500
  101 L=L+1                                                             RRR07510
      IF (L.EQ.NISV(M)) GO TO 110                                       RRR07520
```

```
  102 LL=L+1
      IF (IDATAT(L).LE.IDATAT(LL)) GO TO 101
      LTEMP = IDATAT(L)
      IDATAT(L) = IDATAT(LL)
      IDATAT(LL) = LTEMP
      IF (L.EQ.1) GO TO 102
      L=L-1
      GO TO 102
  110 CONTINUE
C     WRITE OUT THE IDENTIFICATIONS
      WRITE (6,904) (IDATAT(K),K=1,LSTNO)
      WRITE(7,920)( IDATAT(K),K=1,LSTNO)
920   FORMAT(16I5)
  115 CONTINUE
C     WRITE OUT THE WITHIN-CLUSTERS MATRIX
  119 WRITE (6,905)
      DO 120 J=1,NVARS
      WRITE (6,906) (W(K,J),K=1,J)
  120 CONTINUE
C     ICRIT IS THE CRITERION CHOSEN BY THE USER
      GO TO (150,160,170,170),ICRIT
C     WRITE OUT TRACE W
  150 WRITE (6,907) CRIT
      RETURN
C     WRITE OUT DET W
C     AND LOG (DET T / DET W)
  160 CALL UPRFCT (NVARS,T,M)
      DET = 1.
      DO 165 J=1,NVARS
  165 DET = DET*T(J,J)
      WRITE (6,912) CRIT
      CRIT = ALOG 10 ((DET**2)/(CRIT**2))
      WRITE (6,909) CRIT
      RETURN
C     FOR LARGEST ROOT AND HOTELLING'S TRACE CRITERIA, FIND EIGENVALUES
C     FOR |B-GW|=0.
  170 DO 175 J=1,NVARS
      DO 175 K=J,NVARS
      BT(J,K) = B(J,K)
  175 BT(K,J) = BT(J,K)
      CRIT = 1.0 / CRIT
      CALL UTISUI (NVARS,BT,WFCT)
      CALL EIGN (NVARS,BT,EIG,VEC,IND)
      IF (IND.NE.0) GO TO 180
C     WRITE OUT EIGENVALUES
      WRITE (6,910) (EIG(J),J=1,NVARS)
      IF (ICRIT.EQ.4) GO TO 178
C     WRITE OUT LARGEST ROOT
```

```
      WRITE (6,913) CRIT                                                RRR07990
      RETURN                                                            RRR08000
C    WRITE OUT SUM OF EIGENVALUES                                       RRR08010
  178 WRITE (6,914) CRIT                                                RRR08020
      RETURN                                                            RRR08030
C    ESCAPE ROUTE                                                       RRR08040
  180 IFINE = 1                                                         RRR08050
      WRITE (6,911) IND                                                 RRR08060
      RETURN                                                            RRR08070
C                                                                       RRR08080
C     FORMAT STATEMENTS                                                 RRR08090
C                                                                       RRR08100
  900 FCRMAT ('OCLUSTER ',I2)                                           RRR08110
  901 FORMAT ('        SIZE = ',I3)                                     RRR08120
  902 FCRMAT ('        CENTER = ',10(F9.3,1X))                          RRR08130
  903 FORMAT ('        THE OBSERVATIONS ARE ')                          RRR08140
904   FORMAT(8X,10I7)
  905 FORMAT ('OTHE WITHIN CELLS MATRIX IS ')                           RRR08160
  906 FORMAT ('        ',10(F10.3,1X))                                  RRR08170
  907 FORMAT ('OTHE TRACE OF THE WITHIN-CLUSTERS MATRIX IS ',E12.6)     RRR08180
  908 FCRMAT ('O DETERMINANT T IS ',E12.6)                              RRR08190
  909 FORMAT ('O LOG (DET T / DET W) = ',E12.6)                         RRR08200
  910 FORMAT ('O THE EIGENVALUES OF W(INVERSE)*B ARE ',10F8.3)          RRR08210
  911 FORMAT ('1 ILL-CONDITIONED W MATRIX -- IND IS',I3)                RRR08220
  912 FORMAT('OTHE DETERMINANT OF THE WITHIN-CLUSTERS MATRIX IS ',E12.6)RRR08230
  913 FCRMAT ('OTHE LARGEST ROOT IS ',E12.6)                            RRR08240
  914 FORMAT ('OTHE TRACE OF W(INV)*B IS ',E12.6)                       RRR08250
      END                                                               RRR08260
      SUBROUTINE DISTCE (XVECS,YVECS,DIST)                              RRR08270
C                                                                       RRR08280
C     THIS SUBROUTINE CALCULATES EUCLIDIAN, WEIGHTED EUCLIDIAN, OR      RRR08290
C     MAHALANOBIS DISTANCE (DEPENDING ON IDIST).                        RRR08300
C                                                                       RRR08310
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR08320
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR08330
     2ICATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20), RRR08340
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20),NS(600),
     4TOP(20,20),BOT(20),UP(20,20),DOWN(20)
      DIMENSION X(20),XVECS(1),YVECS(1)                                 RRR08360
      IDIR = IDIST+1                                                    RRR08370
      GO TO (100,200,300),IDIR                                          RRR08380
      WRITE (6,900)                                                     RRR08390
  900 FORMAT ('1 ERROR IN DISTANCE CODE - COLUMN 10 OF PROBLEM CARD')   RRR08400
      IFINE = 1                                                         RRR08410
      RETURN                                                            RRR08420
C                                                                       RRR08430
C     CALCULATE EUCLIDIAN DISTANCE                                      RRR08440
C                                                                       RRR08450
```

```
  100 DIST=0.                                                           RRR08460
      DO 150  J=1,NVARS                                                 RRR08470
  150 DIST = DIST + (XVECS(J)-YVECS(J))**2                              RRR08480
      RETURN                                                            RRR08490
C                                                                       RRR08500
C     CALCULATE EUCLIDIAN DISTANCE FOR STANDARDIZED VARIABLES           RRR08510
C                                                                       RRR08520
  200 DIST=0.                                                           RRR08530
      DO 205  J=1,NVARS                                                 RRR08540
  205 DIST = DIST + (XVECS(J)-YVECS(J))*(1.0/W(J,J))*(XVECS(J)-YVECS(J))RRR08550
      RETURN                                                            RRR08560
C                                                                       RRR08570
C    CALCULATE MAHALANOBIS DISTANCE AS THE ELEMENTS OF                  RRR08580
C    L'(INV)(XVEC-YVEC) SQUARED WHERE L IS THE CHOLESKY FACTOR OF W     RRR08590
C                                                                       RRR08600
  300 DIST=0.                                                           RRR08610
      DO 305  J=1,NVARS                                                 RRR08620
  305 X(J) = XVECS(J) - YVECS(J)                                        RRR08630
      CALL UTIRT (NVARS,1,WFCT,X)                                       RRR08640
      DO 310  J=1,NVARS                                                 RRR08650
  310 DIST = DIST + X(J)**2                                             RRR08660
      RETURN                                                            RRR08670
      END                                                               RRR08680
      SUBROUTINE CRITON (CRIT)                                          RRR08690
C                                                                       RRR08700
C     THIS SUBROUTINE CALCULATES THE CRITERIA VALUES                    RRR08710
C                                                                       RRR08720
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR08730
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR08740
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20), RRR08750
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20),NS(600),
     4TOP(20,20),BCT(20),UP(20,20),DOWN(20)
      GO TO (100,200,300,400),ICRIT                                     RRR08770
      WRITE (6,900)                                                     RRR08780
  900 FORMAT ('1 ERROR IN CRITERION CODE, COLUMN 8 IN PROBLEM CARD.')   RRR08790
      IFINE =1                                                          RRR08800
      RETURN                                                            RRR08810
C                                                                       RRR08820
C     CALCULATE TRACE W                                                 RRR08830
C                                                                       RRR08840
  100 CRIT = 0.                                                         RRR08850
      DO 105 J=1,NVARS                                                  RRR08860
  105 CRIT = CRIT + W(J,J)                                              RRR08870
      RETURN                                                            RRR08880
C                                                                       RRR08890
C     CALCULATE DET W AS THE PRODUCT OF DIAGONAL ELEMENTS OF THE        RRR08900
C     CHOSESKY FACTOR OF W                                              RRR08910
C                                                                       RRR08920
```

```
  200 DET = 1.                                                          RRR08930
      DO 205 J=1,NVARS                                                  RRR08940
  205 DET = DET*WFCT(J,J)                                               RRR08950
      CRIT = DET                                                        RRR08960
      RETURN                                                            RRR08970
C                                                                       RRR08980
C     CALCULATE LARGEST ROOT AS L(INV)*B*L'(INV) WHERE L IS THE         RRR08990
C     CHOLESKY FACTOR OF W                                              RRR09000
C     CRITERION IS RECIPROCAL OF LARGEST ROOT                           RRR09010
C                                                                       RRR09020
  300 DO 305 J=1,NVARS                                                  RRR09030
      DO 305 K=J,NVARS                                                  RRR09040
      BT(J,K) = B(J,K)                                                  RRR09050
  305 BT(K,J) = BT(J,K)                                                 RRR09060
      CALL UTISUI (NVARS,BT,WFCT)                                       RRR09070
      CALL EIGN (NVARS,BT,EIG,VEC,IND)                                  RRR09080
      CRIT = 1.0 / EIG(1)                                               RRR09090
      RETURN                                                            RRR09100
C                                                                       RRR09110
C     CALCULATE HOTELLING'S TRACE AS SUM OF DIAGONAL ELEMENTS OF        RRR09120
C     L(INV)*B*L'(INV)                                                  RRR09130
C     CRITERION IS RECIPROCAL OF THIS SUM                               RRR09140
C                                                                       RRR09150
  400 DO 405 J=1,NVARS                                                  RRR09160
      DO 405 K=J,NVARS                                                  RRR09170
      BT(J,K) = B(J,K)                                                  RRR09180
  405 BT(K,J) = BT(J,K)                                                 RRR09190
      CALL UTISUI (NVARS,BT,WFCT)                                       RRR09200
      CRIT = 0.                                                         RRR09210
      DO 410 J=1,NVARS                                                  RRR09220
  410 CRIT = CRIT+BT(J,J)                                               RRR09230
      CRIT=1.0/ CRIT                                                    RRR09240
      RETURN                                                            RRR09250
      END                                                               RRR09260
      SUBROUTINE WCALC (SV,NI,NGP,IDIR)                                 RRR09270
C                                                                       RRR09280
C     THIS SUBROUTINE CALCULATES THE WITHIN-CELLS MATRIX AND, IF        RRR09290
C     NECESSARY, THE CHOLESKY FACTOR OF THE WITHIN-CELLS MATRIX         RRR09300
C                                                                       RRR09310
      COMMON NOBS,NVARS,NGPS,ICRIT,NOSTAN,IDIST,IFINE,KTIME,IDENT(600), RRR09320
     1DATA(600,20),T(20,20),B(20,20),W(20,20),WFCT(20,20),SVCEN(20,20), RRR09330
     2IDATA(600),NISV(20),VMEAN(20),SD(20),XVEC(20),YVEC(20),BT(20,20), RRR09340
     3 NISVT(20),SVCENT(20,20),IDATAT(600),VEC(20,20),EIG(20),NS(600),
     4TCP(20,20),BOT(20),UP(20,20),DOWN(20)
      DIMENSION SV(20,20),NI(20)                                        RRR09360
C                                                                       RRR09370
C     CALCULATE B AND THEN W = T - B                                    RRR09380
C     B = SUM NISV(M)*SVCEN(M)**2                                       RRR09390
```

```
C                                                                       RRR09400
      DC 100 J=1,NVARS                                                  RRR09410
      DO 100 K=J,NVARS                                                  RRR09420
  100 B(J,K) = 0.                                                       RRR09430
      DO 105 M=1,NGP                                                    RRR09440
      DO 105 J=1,NVARS                                                  RRR09450
      DO 105 K=J,NVARS                                                  RRR09460
      BT(J,K) = SV(M,J) * SV(M,K)                                       RRR09470
105   B(J,K)=B(J,K)+BT(J,K)*BOT(M)
      DO 110 J=1,NVARS                                                  RRR09490
      DO 110 K=J,NVARS                                                  RRR09500
  110 W(J,K) = T(J,K) - B(J,K)                                          RRR09510
C                                                                       RRR09520
C     CALCULATE CHOLESKY FACTOR OF W IF IDIR = 2                        RRR09530
C                                                                       RRR09540
      GO TO (120,115),IDIR                                              RRR09550
  115 DO 116 J=1,NVARS                                                  RRR09560
      DO 116 K=J,NVARS                                                  RRR09570
  116 WFCT(J,K) = W(J,K)                                                RRR09580
      CALL UPRFCT (NVARS,WFCT,M)                                        RRR09590
      IF (M.NE.0) GO TO 125                                             RRR09600
  120 RETURN                                                            RRR09610
  125 IFINE=1                                                           RRR09620
      WRITE (6,900)                                                     RRR09630
  900 FORMAT ('1 THE WITHIN GROUPS MATRIX IS SINGLLAR ')                RRR09640
      RETURN                                                            RRR09650
      END                                                               RRR09660
      FUNCTION URAND(IRAND)                                             RRR09670
C                                                                       RRR09680
C     THIS FUNCTION CALCULATES UNIFORMLY DISTRIBUTED RANDOM NUMBERS.    RRR09690
C     BETWEEN 0 AND 1                                                   RRR09700
C     3**19 CONGRLENTIAL UNIFORM RANDOM NUMBER GENERATOR                RRR09710
C                                                                       RRR09720
      IRAND = IRAND*1162261467                                          RRR09730
      IF (IRAND.GT.0) GO TO 3                                           RRR09740
1     IRAND = -IRAND                                                    RRR09750
3     URAND = FLOAT(IRAND)*0.4656612873E-9                              RRR09760
      RETURN                                                            RRR09770
      END                                                               RRR09780
      SUBROUTINE UPRFCT(N,A,M)                                          RRR09790
C                                                                       RRR09800
C     REPLACE UPPER TRIANGLE OF A SQUARE POSITIVE DEFINITE MATRIX A     RRR09810
C     BY ITS CHOLESKI FACTOR                                            RRR09820
C                                                                       RRR09830
      DIMENSION A(20,20)                                                RRR09840
C     CLEAR THE ERROR INDICATOR                                         RRR09850
      M=0                                                               RRR09860
      N1=N-1                                                            RRR09870
```

```
      IF(N1) 230,100,100                                                 RRR09880
  100 DO 220  K=1,N                                                      RRR09890
      AKK=A(K,K)                                                         RRR09900
      IF(K .EQ. 1) GO TO 120                                             RRR09910
      DO 110  J=2,K                                                      RRR09920
  110 AKK=AKK-A(J-1,K)**2                                                RRR09930
  120 IF(A(K,K)) 140,140,130                                             RRR09940
C     IN THE CASE OF A COVARIANCE MATRIX AKK/A(K,K) IS 1-R**2 WHERE      RRR09950
C     R IS MULT CORRELATION OF VARIABLE K WITH ALL PRECEEDING VARIABLES. RRR09960
  130 IF(AKK/A(K,K) .GE.   .001) GO TO 150                               RRR09970
  140 M=K                                                                RRR09980
      AKK=0.                                                             RRR09990
  150 AKK=SQRT(AKK)                                                      RRR10000
      A(K,K)=AKK                                                         RRR10010
      IF(K .EQ. N) GO TO 230                                             RRR10020
C                                                                        RRR10030
      DO 220 I=K,N1                                                      RRR10040
      AKI=A(K,I+1)                                                       RRR10050
      IF(K .EQ. 1) GO TO 190                                             RRR10060
      DO 180 J=2,K                                                       RRR10070
  180 AKI=AKI-A(J-1,K)*A(J-1,I+1)                                        RRR10080
  190 IF(AKK) 210,200,210                                                RRR10090
  200 A(K,I+1)= 0.0                                                      RRR10100
      GO TO 220                                                          RRR10110
  210 A(K,I+1)=AKI/AKK                                                   RRR10120
  220 CONTINUE                                                           RRR10130
  230 RETURN                                                             RRR10140
      END                                                                RRR10150
      SUBROUTINE UTIRT(M,N,S,B)                                          RRR10160
C                                                                        RRR10170
C     INVERSE OF UPPER (S) TRANSPOSED TIMES RECTANGLE B TRANSPOSED.      RRR10180
C                                                                        RRR10190
      DIMENSION S(20,20),B(1,20)                                         RRR10200
      DO 130 J=1,N                                                       RRR10210
      DO 130 I=1,M                                                       RRR10220
      SUM = 0.0                                                          RRR10230
      IF (S(I,I) ) 90,130,90                                             RRR10240
   90 SUM = B(J,I)                                                       RRR10250
      IM1 = I-1                                                          RRR10260
      IF(IM1) 120,120,100                                                RRR10270
  100 DO 110 K=1,IM1                                                     RRR10280
  110 SUM = SUM-S(K,I)*B(J,K)                                            RRR10290
  120 SUM = SUM/S(I,I)                                                   RRR10300
  130 B(J,I) = SUM                                                       RRR10310
      RETURN                                                             RRR10320
      END                                                                RRR10330
      SUBROUTINE LTISUI (N,A,B)                                          RRR10340
C                                                                        RRR10350
```

```
C      UPPER (B) TRANSPOSE INVERSE TIMES A TIMES UPPER (B) INVERSE.         RRR10360
C                                                                           RRR10370
       DIMENSION A(20,20),B(20,20)                                          RRR10380
C      NOTE THAT POSTMULT IS CARRIED OUT ON FINAL VALUES LEFT BY PREMULT    RRR10390
       DO 200 I=1,N                                                         RRR10400
       I1 = I-1                                                             RRR10410
C      NOTE THAT PREMULT ON RIGHT HALF OF ROW IS SAME AS POSTMULT           RRR10420
C      ON LOWER HALF OF COLUMN - EXCEPT FOR DIAG TERM WHICH IS THE          RRR10430
C      FINAL VALUE LEFT BY PREMULT BEFORE POSTMULT                          RRR10440
       DO 130 J=I,N                                                         RRR10450
       IF(I1) 120,120,100                                                   RRR10460
   100 DO 110 K=1,I1                                                        RRR10470
   110 A(J,I) = A(J,I) - B(K,I)*A(J,K)                                      RRR10480
   120 A(J,I) = A(J,I)/B(I,I)                                               RRR10490
   130 IF (B(I,I) .EQ. 0.0) A(J,I) = 0.0                                    RRR10500
C      NOTE THAT ELEMENTS IN LEFT HALF OF ROW ARE FINAL FOR PREMULT         RRR10510
C      NOTE THAT DIAG ELEMENT WAS PREVIOUSLY THE FINAL RESULT OF            RRR10520
C      PREMULT.  NOW WE MAKE THE FINAL RESULT OF POSTMULT.                  RRR10530
       DO 200 J=1,I                                                         RRR10540
       J1 = J-1                                                             RRR10550
       IF(J1) 160,160,140                                                   RRR10560
C      HORIZONTAL BRANCH OF INNER PRODUCT EXCLUDING DIAG TERM               RRR10570
   140 DO 150 K=1,J1                                                        RRR10580
   150 A(I,J) = A(I,J) - B(K,I)*A(J,K)                                      RRR10590
   160 IF(I1-J) 190,170,170                                                 RRR10600
C      VERTICAL BRANCH OF INNER PRODUCT INCLUDING DIAG TERM                 RRR10610
   170 DO 180 K=J,I1                                                        RRR10620
   180 A(I,J) = A(I,J) - B(K,I)*A(K,J)                                      RRR10630
   190 A(I,J) = A(I,J)/B(I,I)                                               RRR10640
   200 IF (B(I,I) .EQ. 0.0) A(I,J) = 0.0                                    RRR10650
       RETURN                                                               RRR10660
       END                                                                  RRR10670
       SUBROUTINE EIGN(NN,A,EIG,VEC,IND)                                    RRR10680
C          NN= SIZE OF MATRIX                                               RRR10690
C          A = MATRIX (ONLY LOWER TRIANGLE IS USED + THIS IS DESTROYED)     RRR10700
C          EIG = RETURNED EIGENVALUES IN ALGEBRAIC DESCENDING ORDER         RRR10710
C          VEC = RETURNED EIGENVECTORS IN COLUMNS                           RRR10720
C          IND = ERROR RETURN INDICATOR                                     RRR10730
C                0 FOR NORMAL RETURN                                        RRR10740
C                1 SUM OF EIGENVALUES NOT EQUAL TO TRACE                    RRR10750
C                2 SUM OF EIGENVALUES SQUARED NOT EQUAL TO NORM             RRR10760
C                3 BOTH OF THESE ERRORS                                     RRR10770
       DIMENSION A(20,20),GAMMA(20),BETA(20),BETASQ(20),EIG(20)             RRR10780
       DIMENSION W(20),VEC(20,20)                                           RRR10790
C      THE FOLLOWING DIMENSIONED VARIABLES ARE EQUIVALENCED                 RRR10800
       DIMENSION P(19),Q(19)                                                RRR10810
       EQUIVALENCE (P(1),BETA(1)),(Q(1),BETA(1))                            RRR10820
       DIMENSION IPOSV(20),IVPOS(20),IORD(20)                               RRR10830
```

```
      ECUIVALENCE (IPOSV(1),GAMMA(1)),(IVPOS(1),BETA(1)),              RRR10840
     1(IORD(1),BETASQ(1))                                              RRR10850
      N=NN                                                             RRR10860
C     RESET ERROR RETURN INDICATOR                                     RRR10870
      IND=0                                                            RRR10880
      IF(N .EQ. 0) GO TO 560                                           RRR1C890
      N1=N-1                                                           RRR10900
      N2=N-2                                                           RRR10910
C     COMPUTE THE TRACE AND EUCLIDIAN NCRM OF THE INPUT MATRIX         RRR10920
C     LATER CHECK AGAINST SUM AND SUM OF SQUARES OF EIGENVALUES        RRR10930
      ENCRM=0.
      TRACE=0.                                                         RRR10950
      DC 110 J=1,N                                                     RRR10960
      DO 100 I=J,N                                                     RRR10970
  100 ENORM=ENORM+A(I,J)**2                                            RRR10980
      TRACE=TRACE+A(J,J)                                               RRR10990
  110 ENORM=ENORM-.5*A(J,J)**2                                         RRR11C00
      ENORM=ENORM+ENORM                                                RRR11010
      GAMMA(1)=A(1,1)                                                  RRR11020
      IF(N2) 280,270,120                                               RRR11030
  120 DO 260 NR=1,N2                                                   RRR11040
      B=A(NR+1,NR)                                                     RRR11050
      S=0.                                                             RRR11060
      DO 130 I=NR,N2                                                   RRR11070
  130 S=S+A(I+2,NR)**2                                                 RRR11080
C     PREPARE FOR POSSIBLE BYPASS OF TRANSFORMATICN                    RRR11090
      A(NR+1,NR)=0.                                                    RRR11100
      IF(S) 250,250,140                                                RRR11110
  140 S=S+B*B                                                          RRR11120
      SGN=+1.                                                          RRR11130
      IF(B) 150,160,160                                                RRR11140
  150 SGN=-1.                                                          RRR11150
  160 SCRTS=SQRT(S)                                                    RRR11160
      D=SGN/(SQRTS+SQRTS)                                              RRR11170
      TEMP=SQRT(.5+B*D)                                                RRR11180
      W(NR)=TEMP                                                       RRR11190
      A(NR+1,NR)=TEMP                                                  RRR11200
      D=D/TEMP                                                         RRR11210
      B=-SGN*SQRTS                                                     RRR11220
C     D IS FACTOR OF PROPORTIONALITY. NOW COMPUTE AND SAVE W VECTOR.   RRR11230
C     EXTRA SINGLY SUBSCRIPTED W VECTOR USED FOR SPEED.                RRR11240
      DO 170 I=NR,N2                                                   RRR11250
      TEMP=D*A(I+2,NR)                                                 RRR11260
      W(I+1)=TEMP                                                      RRR11270
  17C A(I+2,NR)=TEMP                                                   RRR11280
C     PREMULTIPLY VECTOR W BY MATRIX A TO OBTAIN P VECTOR.             RRR11290
C     SIMULTANEOUSLY ACCUMULATE DOT PRODUCT WP,(THE SCALAR K)          RRR11300
      WTAW=0.                                                          RRR11310
```

```
      DC 220 I=NR,N1                                                   RRR11320
      SUM=0.                                                           RRR11330
      CC 180 J=NR,I                                                    RRR11340
  180 SUM=SUM+A(I+1,J+1)*W(J)                                          RRR11350
      I1=I+1                                                           RRR11360
      IF(N1-I1) 210,190,190                                            RRR11370
  19C DO 200 J=I1,N1                                                   RRR11380
  200 SUM=SUM+A(J+1,I+1)*W(J)                                          RRR11390
  210 P(I)=SUM                                                         RRR11400
  22C WTAW=WTAW+SUM*W(I)                                               RRR11410
C     P VECTOR AND SCALAR K NOW STORED. NEXT COMPLTE Q VECTOR          RRR11420
      DO 230 I=NR,N1                                                   RRR11430
  230 Q(I)=P(I)-WTAW*W(I)                                              RRR11440
C     NOW FORM PAP MATRIX, REQUIRED PART                               RRR11450
      DO 240 J=NR,N1                                                   RRR11460
      QJ=Q(J)                                                          RRR11470
      WJ=W(J)                                                          RRR11480
      DO 240 I=J,N1                                                    RRR11490
  240 A(I+1,J+1)=A(I+1,J+1)-2.*(W(I)*QJ+WJ*Q(I))                       RRR11500
  250 BETA(NR)=B                                                       RRR11510
      BETASQ(NR)=B*B                                                   RRR11520
  260 GAMMA(NR+1)=A(NR+1,NR+1)                                         RRR11530
  27C B=A(N,N-1)                                                       RRR11540
      BETA(N-1)=B                                                      RRR11550
      BETASQ(N-1)=B*B                                                  RRR11560
      GAMMA(N)=A(N,N)                                                  RRR11570
  280 BETASQ(N)=0.                                                     RRR11580
C     ADJOIN AN ICENTITY MATRIX TO BE POSTMULTIPLIED BY ROTATIONS.     RRR11590
      DO 300 I=1,N                                                     RRR11600
      CO 290 J=1,N                                                     RRR11610
  290 VEC(I,J)=0.                                                      RRR11620
  300 VEC(I,I)=1.                                                      RRR11630
      M=N                                                              RRR11640
      SUM=0.                                                           RRR11650
      NPAS=1                                                           RRR11660
      GO TO 400                                                        RRR11670
  31C SUM=SUM+SHIFT                                                    RRR11680
      COSA=1.                                                          RRR11690
      G=GAMMA(1)-SHIFT                                                 RRR11700
      PP=G                                                             RRR11710
      PPBS=PP*PP+BETASQ(1)                                             RRR11720
      PPBR=SQRT(PPBS)                                                  RRR11730
      DO 370 J=1,M                                                     RRR11740
      CCSAP=COSA                                                       RRR11750
      IF(PPBS .NE. 0.) GO TO 320                                       RRR11760
      SINA=0.                                                          RRR11770
      SINA2=0.                                                         RRR11780
      COSA=1.                                                          RRR11790
```

```
      GO TO 350
  32C SINA=BETA(J)/PPBR
      SINA2=BETASQ(J)/PPBS
      COSA=PP/PPBR
C     POSTMULTIPLY IDENTITY BY P-TRANSPCSE MATRIX
      NT=J+NPAS
      IF(NT .GE. N) NT=N
  330 DO 340 I=1,NT
      TEMP=COSA*VEC(I,J)+SINA*VEC(I,J+1)
      VEC(I,J+1)=-SINA*VEC(I,J)+COSA*VEC(I,J+1)
  340 VEC(I,J)=TEMP
  350 CIA=GAMMA(J+1)-SHIFT
      U=SINA2*(G+DIA)
      GAMMA(J)=G+U
      G=DIA-U
      PP=DIA*COSA-SINA*COSAP*BETA(J)
      IF(J .NE. M) GO TO 360
      BETA(J)=SINA*PP
      BETASQ(J)=SINA2*PP*PP
      GC TO 380
  360 PFBS=PP*PP+BETASQ(J+1)
      PPBR=SQRT(PPBS)
      BETA(J)=SINA*PPBR
  37C BETASQ(J)=SINA2*PPBS
  380 GAMMA(M+1)=G
C     TEST FOR CONVERGENCE OF LAST DIAGONAL ELEMENT
      NPAS=NPAS+1
      IF(BETASQ(M) .GT. 1.E-21) GO TO 41C
  390 EIG(M+1)=GAMMA(M+1)+SUM
  400 BETA(M)=0.
      BETASQ(M)=0.
      M=M-1
      IF(M .EQ. 0) GO TO 430
      IF(BETASQ(M) .LE. 1.E-21) GO TO 390
C     TAKE ROOT OF CORNER 2 BY 2 NEAREST TO LOWER DIAGONAL IN VALUE
C     AS ESTIMATE OF EIGENVALUE TO USE FOR SHIFT
  410 A2=GAMMA(M+1)
      R2=.5*A2
      R1=.5*GAMMA(M)
      R12=R1+R2
      DIF=R1-R2
      TEMP=SQRT(DIF*DIF+BETASQ(M))
      R1=R12+TEMP
      R2=R12-TEMP
      DIF=ABS(A2-R1)-ABS(A2-R2)
      IF(DIF .LT. 0.) GO TO 420
      SHIFT=R2
      GO TO 310
```

```
  420 SHIFT=R1                                                          RRR12280
      GO TO 310                                                         RRR12290
  430 EIG(1)=GAMMA(1)+SUM                                               RRR12300
C     INITIALIZE AUXILIARY TABLES REQUIRED FOR REARRANGING THE VECTORS  RRR12310
      DO 440 J=1,N                                                      RRR12320
      IPOSV(J)=J                                                        RRR12330
      IVPOS(J)=J                                                        RRR12340
  440 IORD(J)=J                                                         RRR12350
C     USE A TRANSPOSITION SORT TO ORDER THE EIGENVALUES                 RRR12360
      M=N                                                               RRR12370
      GO TO 470                                                         RRR12380
  450 DO 460 J=1,M                                                      RRR12390
      IF(EIG(J) .GE. EIG(J+1)) GO TO 460                                RRR12400
      TEMP=EIG(J)                                                       RRR12410
      EIG(J)= EIG(J+1)                                                  RRR12420
      EIG(J+1)=TEMP                                                     RRR12430
      ITEMP=IORD(J)                                                     RRR12440
      IORD(J)=IORD(J+1)                                                 RRR12450
      IORD(J+1)=ITEMP                                                   RRR12460
  460 CONTINUE                                                          RRR12470
  470 M=M-1                                                             RRR12480
      IF(M .NE. 0) GO TO 450                                            RRR12490
      IF(N1 .EQ. 0) GO TO 500                                           RRR12500
      DO 490 L=1,N1                                                     RRR12510
      NV=IORD(L)                                                        RRR12520
      NP=IPOSV(NV)                                                      RRR12530
      IF(NP .EQ. L) GO TO 490                                           RRR12540
      LV=IVPOS(L)                                                       RRR12550
      IVPOS(NP)=LV                                                      RRR12560
      IPOSV(LV)=NP                                                      RRR12570
      DO 480 I=1,N                                                      RRR12580
      TEMP=VEC(I,L)                                                     RRR12590
      VEC(I,L)=VEC(I,NP)                                                RRR12600
  480 VEC(I,NP)=TEMP                                                    RRR12610
  490 CONTINUE                                                          RRR12620
  500 ESUM=0.                                                           RRR12630
      ESSQ=0.                                                           RRR12640
C     BACK TRANSFORM THE VECTORS OF THE TRIPLE DIAGONAL MATRIX          RRR12650
      DO 550 NRR=1,N                                                    RRR12660
      K=N1                                                              RRR12670
  510 K=K-1                                                             RRR12680
      IF(K .LE. 0) GO TO 540                                            RRR12690
      SUM=0.                                                            RRR12700
      DO 520 I=K,N1                                                     RRR12710
  520 SUM=SUM+VEC(I+1,NRR)*A(I+1,K)                                     RRR12720
      SUM=SUM+SUM                                                       RRR12730
      DO 530 I=K,N1                                                     RRR12740
  530 VEC(I+1,NRR)=VEC(I+1,NRR)-SUM*A(I+1,K)                            RRR12750
```

```
      GO TO 510                                                         RRR12760
  540 ESUM=ESUM+EIG(NRR)                                                RRR12770
  550 ESSQ=ESSQ+EIG(NRR)**2                                             RRR12780
      TEMP=ABS(512.*TRACE)                                              RRR12790
      IF((ABS(TRACE-ESUM)+TEMP)-TEMP .NE. 0.) IND=IND+1                 RRR12800
      TEMP=1024.*ENORM                                                  RRR12810
      IF((ABS(ENORM-ESSQ)+TEMP)-TEMP .NE. 0.) IND=IND+2                 RRR12820
  560 RETURN                                                            RRR12830
      END                                                               RRR12840
//GO.SYSIN DD *
  5 14 11 15 22  8 16 15 12 21  9 12 12 27  4 18  9 12 20 21  9  8 25 16  3
 12 16 14  2  2  6 28 36 25 15 29  9 14 17 16 28 10 12 12 11 15 12 13 13 36
 16 18  8  3 23 14 21 15 10  4 24  9  6 20 30 25 37  9 25 12 17 14 24 15 15
 19 10  5  7  9  8 19 23 17 10 24 24 19 21 20  8  2  3  8  2  7  3  9  9  7
  9  2  2  3  6  3 20  6  7  8 15 19  7  9  7  7 22 17  9 12  8 15 12  3 12
  8  9 12 35 19 33 21  9 26 21 20 15 21 39 11  2  6  9 11 26  8 15  8 11 12
  2  5  6  6  9 13  7  2  6  9 13  7  3  4  6  8  3 10  3 10 17  6 18  8 15
 12  6  2  7  5 13  6  9  6 10 18  8  5  9  9
 WEIGHTED  773            TEST FOR CPU TIME
 190141021209
(I3,5X,7F8.4,/8X,7F8.4)
```

```
//AIK$C$79 JOB (1642,0526,RJ74),'AIKEN SMC 2132'
// EXEC FORTCLG,REGION.GO=180K
//FORT.SYSIN DD *
C
C
C
C     THIS METHOD OF COMPARING CLUSTERS OF OBJECTS BY MULTIPLE JUDGES
C     WAS DESIGNED BY JOEL AIKEN WHO WAS GIVEN THE IDEA BY PROF J. HARTMAN.
C
C
C
C     ARRANGE DATA DECK SO THAT CARDS ARE IN THIS ORDER:
C     TITLE CARD FOLLOWED BY:
C     1.  # OBJECTS, # JUDGES (2I3)
C     2.  # OF CLUSTERS BY FIRST JUDGE  (I3)
C     3.  # OBJECTS IN EACH CLUSTER  (15I5)
C        (NOTE: 15 CLUSTERS IS MAX)
C     4.  OBJECT ID NUMBERS FOR CLUSTERS  (16I5)
C         REPEAT AS NECESSARY, 2 CARDS PER CLUSTER.
C     5.  RETURN TO STEP 2 FOR SUBSEQUENT JUDGES.
C
C     THE PROGRAM COMPUTES A SUM OF SQUARES VALUE AND DIVIDES THE SS
C     BY THE BEST POSSIBLE SS AND THIS RATIO IS CALLED 'C'.
C     THE NUMBER AND SIZE OF CLUSTERS DETERMINE THE BEST SS.
C     CSTAR IS COMPUTED AS  (SS-LOW)/(BEST-LOW)
C
C     THE UPPER BOUND IS FOUND BY CONSIDERING THE # OF JUDGES,
C     # OF CLUSTERS, AND # OF OBJECTS IN EACH CLUSTER AND FROM THAT
C     DETERMINING THE BEST POSSIBLE MATCH MATRIX WHICH COULD EXIST.
C     THIS PROCEDURE USES A MINIMAX APPROACH AND IS PERFORMED BY
C     SUBROUTINE UPPER.  THE NUMBER C IS THEN COMPUTED AS THE RATIO
C     OF ACTUAL TO BEST SUM OF SQUARES.  THIS PROCEDURE IS AN ATTEMPT TO
C     ALLOW THE USER A STANDARD SCALE UPON WHICH TO COMPARE C VALUES
C     BASED ON VARYING CLUSTER NUMBERS AND SIZES.
C     SEE SUBROUTINE UPPER FOR DETAILS
C
C
C
      COMMON IX1(100,100),IX2(100,15),ITEMP(100,100),NINC1(100),
     1NINC2(15),KCUNT1(100),NCLUS1,NCLUS2
      DIMENSION IVY(50,10),L(10),TITLE(20),JROW(10)
      SUM=0.0
      READ(5,15)(TITLE(I),I=1,20)
15    FORMAT(20A4)
      WRITE(6,16)(TITLE(I),I=1,20)
16    FORMAT(' ',20A4,//)
C
C
```

```
C
C     READ IN TOTAL # OBJECTS TO BE CLUSTERED AND
C     THE NUMBER OF JUDGES WHO WILL DO INDEPENDENT CLUSTERING
C
      READ(5,100) NOBJ,NJUDGE
100   FORMAT(2I3)
      DO 20 J=1,100
      DO 10 I=1,50
      ITEMP(I,J)=0
10    CONTINUE
20    CONTINUE
C
C
C
C     READ IN THE DATA FROM THE FIRST TWO JUDGES
C
      READ(5,105) NCLUS1
105   FORMAT(I3)
      L(1)=NCLUS1
      READ(5,110)(NINC1(I),I=1,NCLUS1)
110   FORMAT(15I5)
      WRITE(6,106) NOBJ,NJUDGE
106   FORMAT('0 NUMBER OF OBJECTS TO BE CLUSTERED IS ',I3,
     110X,'NUMBER OF JUDGES IS ',I3)
      WRITE(6,108)
108   FORMAT('0',////,' JUDGE  1')
      WRITE(6,107) NCLUS1,(NINC1(I),I=1,NCLUS1)
107   FORMAT('0NUMBER OF CLUSTERS IS ',I3,10X,
     1' NUMBER IN EACH CLUSTER IS ',15I5,/)
      MIN=NCLUS1
      PRCD=FLOAT(NCLUS1)
      DO 200 I=1,NCLUS1
      ILIM1=NINC1(I)
C     IVY IS MATRIX WHOSE COLS ARE NINC'S (ONE COL FOR EACH JUDGE)
      IVY(I,1)=NINC1(I)
      READ(5,115)(IX1(J,I),J=1,ILIM1)
115   FORMAT(16I5)
      WRITE(6,116)(IX1(J,I),J=1,ILIM1)
116   FORMAT('0',/25I5,/25I5,/25I5,/25I5)
200   CONTINUE
      NJUDG1=NJUDGE-1
C     MERGE ALL JUDGES
      DO 300 K=1,NJUDG1
      K1=K+1
      READ(5,105) NCLUS2
      L(K1)=NCLUS2
C     MIN AND PROD ARE INFO FOR C-STATISTIC.
      IF(NCLUS2.LT.NCLUS1) MIN=NCLUS2
```

```
      PROD=PROD*FLOAT(NCLUS2)
      READ(5,110)(NINC2(I),I=1,NCLUS2)
      WRITE(6,220) K1
220   FORMAT('0',////,' JUDGE ',I2)
      WRITE(6,107) NCLUS2,(NINC2(I),I=1,NCLUS2)
      DO 250 I=1,NCLUS2
      ILIM2=NINC2(I)
C     FORM ANOTHER COL OF IVY AND READ IN NEXT JUDGE'S DATA.
      IVY(I,K1)=NINC2(I)
      READ(5,115)(IX2(J,I),J=1,ILIM2)
      WRITE(6,116)(IX2(J,I),J=1,ILIM2)
250   CONTINUE
C
C
C
C     MERGE THE FIRST TWO
C     (AND SUBSEQUENTLY, TWO AT A TIME, BUILDING A MERGED CLUSTER SET.
C
      CALL MERGE(IFLAG,ITCOL)
290   FORMAT(' ',/////,' FINAL MERGED MATRIX',/)
300   CONTINUE
      WRITE(6,290)
C     ITCOL IS NUMBER OF MERGED CLUSTERS
      DO 350 I=1,ITCOL
C     KOUNT1 IS VECTOR CONTAINING # OF OBS IN EACH CLUSTER OF THE MERGED SET.
      ILIM3=KOUNT1(I)
      WRITE(6,116)(IX1(J,I),J=1,ILIM3)
      SUM=SUM+(KOUNT1(I))**2
350   CONTINUE
C
C
C
C
C     COMPUTE MEASURE OF SIMILARITY
C
      CALL UPPER(IVY,L,JROW,NJUDGE,U,NOBJ)
      TOP=(FLOAT(NOBJ))**2
      IF(NOBJ.GT.PROD) GO TO 490
      XL=NOBJ
      GO TO 491
490   NG=NOBJ/PROD
      RE=AMOD((FLOAT(NOBJ)),PROD)
      XL=RE*((NG+1)**2)+(PROD-RE)*(NG**2)
491   C=SUM/U
      CSTAR=(SUM-XL)/(U-XL)
      WRITE(6,500) SUM,U,XL,C,CSTAR
500   FORMAT('0',///,' RESULTS',/,' SUM OF SQUARES = ',F10.4,/,
     1' BEST POSSIBLE = ',F10.4,10X,'WORST POSSIBLE = ',F10.4,/,
     2' C = ',F10.4,5X,'(NORMALIZED COEFFICIENT OF SIMILARITY)',
```

```
      3//,' CSTAR = ',F10.4)
       STOP
       ENC
C
       SUBROUTINE MERGE(IFLAG,ITCOL)
       COMMON IX1(100,100),IX2(100,15),ITEMP(100,1C0),NINC1(100),
      1NINC2(15),KCUNT1(100),NCLUS1,NCLUS2
       IFLAG=0
       ITCOL=1
       KCUNT1(1)=1
C      BIG LOOP (4 DEEP)
C      CCMPARES EVERY POSSIBLE PAIR FOR A MATCH.  ORDERED SO THAT COL 1 OF IX2
C      IS CHECKED AGAINST ALL OF IX1(COL AT A TIME), THEN CCL 2 OF IX1, ETC.
       DO 500 N=1,NCLUS2
       LIM4=NINC2(N)
       DO 400 J=1,NCLUS1
       DC 300 M=1,LIM4
       LIM5=NINC1(J)
       DO 200 I=1,LIM5
       IF(IX2(M,N).NE.IX1(I,J)) GO TO 200
C      FOUND A MATCH: TMAT WILL RECORD IT.
       CALL TMAT(IFLAG,IX2(M,N),ITCOL,J,N)
       GC TO 300
200    CONTINUE
300    CONTINUE
400    CONTINUE
C      ALTER WILL SET UP FOR NEXT JUDGE.
       IF(N.EQ.NCLUS2) CALL ALTER(ITCOL)
500    CONTINUE
       RETURN
       ENC
C
       SUBROUTINE TMAT(IFLAG,IVAL,ITCOL,J,N)
       CCMMON IX1(100,100),IX2(100,15),ITEMP(100,1C0),NINC1(10C),
      1NINC2(15),KCUNT1(100),NCLUS1,NCLUS2
C      CHECKS FOR FIRST TIME INTO HERE CN THIS MERGE.
       IF (IFLAG.EQ.0) GO TO 150
C      CHECKS IF COLS ARE STILL THE SAME AS LAST MATCH; IF SO, WILL ADD ANOTHER
C      OBS TO THAT PARTICULAR MERGED CLUSTER.
       IF(ICOL1.EQ.J.AND.ICOL2.EQ.N) GO TO 100
C      IF NEW COL, THEN NEW MERGED CLUSTER.
       ITCOL=ITCOL+1
C      KOUNT1(I) IS # OF OBS IN I-TH MERGED CLUSTER.
       KOUNT1(ITCOL)=1
       GC TO 150
100    KCUNT1(ITCOL)=KOUNT1(ITCOL)+1
150    ITEMP(KOUNT1(ITCOL),ITCOL)=IVAL
C      SET UP FOR NEXT CALL.  THESE WILL BE OLD COL #'S AGAINST WHICH WILL BE
```

```
C       CHECKED THE NEW COL #'S.
        ICOL1=J
        ICOL2=N
        IFLAG=1
        RETURN
        END
C
        SUBROUTINE ALTER(ITCOL)
        COMMON IX1(100,100),IX2(100,15),ITEMP(100,100),NINC1(100),
       1NINC2(15),KOUNT1(100),NCLUS1,NCLUS2
C
C
C
C       MAKES THE NAME OF THE MERGED SET: IX1.
C       SO THE NEXT SET TO BE READ IN WILL BE A NEW IX2.
C
        NCLUS1=ITCOL
        DO 100 I=1,ITCOL
        NINC1(I)=KOUNT1(I)
        LIM6=KOUNT1(I)
        DO 50 J=1,LIM6
        IX1(J,I)=ITEMP(J,I)
50      CONTINUE
100     CONTINUE
        RETURN
        END
C
        SUBROUTINE UPPER(IVY,L,JROW,NJUDGE,BEST,NOBJ)
        DIMENSION IVY(50,10),L(10),JROW(10)
        ISUM=0
        BEST=0.0
C
C
C
C       THE MATRIX IVY IS CONSTRUCTED AS FOLLOWS:
C       COLS = JUDGES
C       ROWS = CLUSTERS
C       THE (I,J)-TH ELEMENT OF THE MATRIX IS THE NUMBER OF OBJECTS IN
C       THE I-TH CLUSTER OF THE J-TH JUDGE.
C
C       FIND MAX IN EACH COLUMN
C
10      DO 100 I=1,NJUDGE
        LIM1=L(I)
        DO 75 J=1,LIM1
        IF(J.NE.1) GO TO 60
        MAX=IVY(J,I)
        JROW(I)=J
```

```
60     IF(IVY(J,I).LE.MAX) GO TO 75
       MAX=IVY(J,I)
       JROW(I)=J
75     CONTINUE
100    CONTINUE
C
C
C      NOW FIND MIN OF MAX'S AND SUBTRACT FROM EACH MAX
C
       DC 200 I=1,NJUDGE
       IF(I.EQ.1) MIN=IVY(JROW(I),I)
       IF (IVY(JROW(I), I).LT.MIN)MIN= IVY(JROW(I),I)
200    CONTINUE
C
C
C      ACCUMULATE SUM OF MINIMAX'S AND SUM OF SQUARES
C
       ISUM=ISUM+MIN
       BEST=BEST+MIN**2
       DO 300 I=1,NJUDGE
       IVY(JROW(I),I)=IVY(JROW(I),I)-MIN
300    CONTINUE
       IF(ISUM.LT.NOBJ) GO TO 10
       RETURN
       END
//GO.SYSIN DD *
```

BIBLIOGRAPHY

1. Anderberg, M.R., Cluster Analysis For Applications, p. 3, Academic Press, 1973.

2. MacQueen, J., Some Methods for Classification and Analysis of Multivariate Observations, paper presented at Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California, 1965-1966.

3. McRae, D.J., Clustering Multivariate Observations, doctoral dissertation, University of North Carolina, Chapel Hill, N.C., 1973.

4. Friedman, H.P., and Rubin, J., "On Some Invariant Criteria for Grouping Data," Journal of the American Statistical Association, v. 62: 320, p. 1159-1178, December 1967.

5. Eisenheis, R.A. and Avery, R.B., Discriminant Analysis and Classification Procedures, p. 37, Lexington Books, 1972.

6. Anderson, T.W., An Introduction to Multivariate Statistical Analysis, Wiley, 1958.

7. Read, R.R., "A Study of SOF Data," unpublished report at Naval Postgraduate School, Monterey, Ca., 1 December 1977.

8. Lindsay, G.F., "On Constructing Interval Scales From Ordinal Judgments," July 1977; "Categorical Judgments: The Method of Successive INtervals", August 1976, unpublished reports at the Naval Postgraduate School; also Memorandum to Professor R.R. Read dated 2 October 1978, [NC4(551s)/pb], subject: "Results of OS 4207 Class Project in Scaling SOF Data."

9. Wilks, S.S., Mathematical Statistics, p. 573-601, Wiley, 1962.

10. Stanford University, Department of Statistics Technical Report No. 71, The Use of Faces to Represent Points in n-Dimensional Space Graphically, by Herman Chernoff, 27 December, 1971.

11. Chernoff, H., "The Use of Faces to Represent Points in k-Dimensional Space Graphically," Journal of the American Statistical Association, V. 68: 342, p. 361-368, June, 1973.

12. Lake, G.E., A Graphical Representation of Multivariate Data: Soviet Foreign Policy in Sub-Saharan Africa, Master's Thesis, Naval Postgraduate School, Monterey, Ca., 1977.

13. Chernoff, H. and Rizvi, M.H., "Error of Random Permutations of Features in Representing Multivariate Data by Faces," Journal of the American Statistical Association, V. 70:351, p. 548-554, September 1975.

14. Siegel, S., Nonparametric Statistics, p. 201, McGraw-Hill, 1956.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4.	Professor R.R. Read, Code 55Re Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
5.	Professor J.R. Borsting, Code 01 Provost Naval Postgraduate School Monterey, California 93940	1
6.	Professor D.E. Kirk, Code 62Ki Chairman Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
7.	LCDR Joel W. Aiken 555 "C" Avenue Coronado, California 92118	1
8.	Scholarship Committee Naval Postgraduate School Monterey, California 93940	1
9.	Professor G.F. Lindsay, Code 55Ls Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
10.	D.J. McRae Cal Test Bureau McGraw-Hill Monterey, California 93940	1